PDP11/70

CACHE DIAGNOSTIC PART 1

MD-11-DEKBC-B

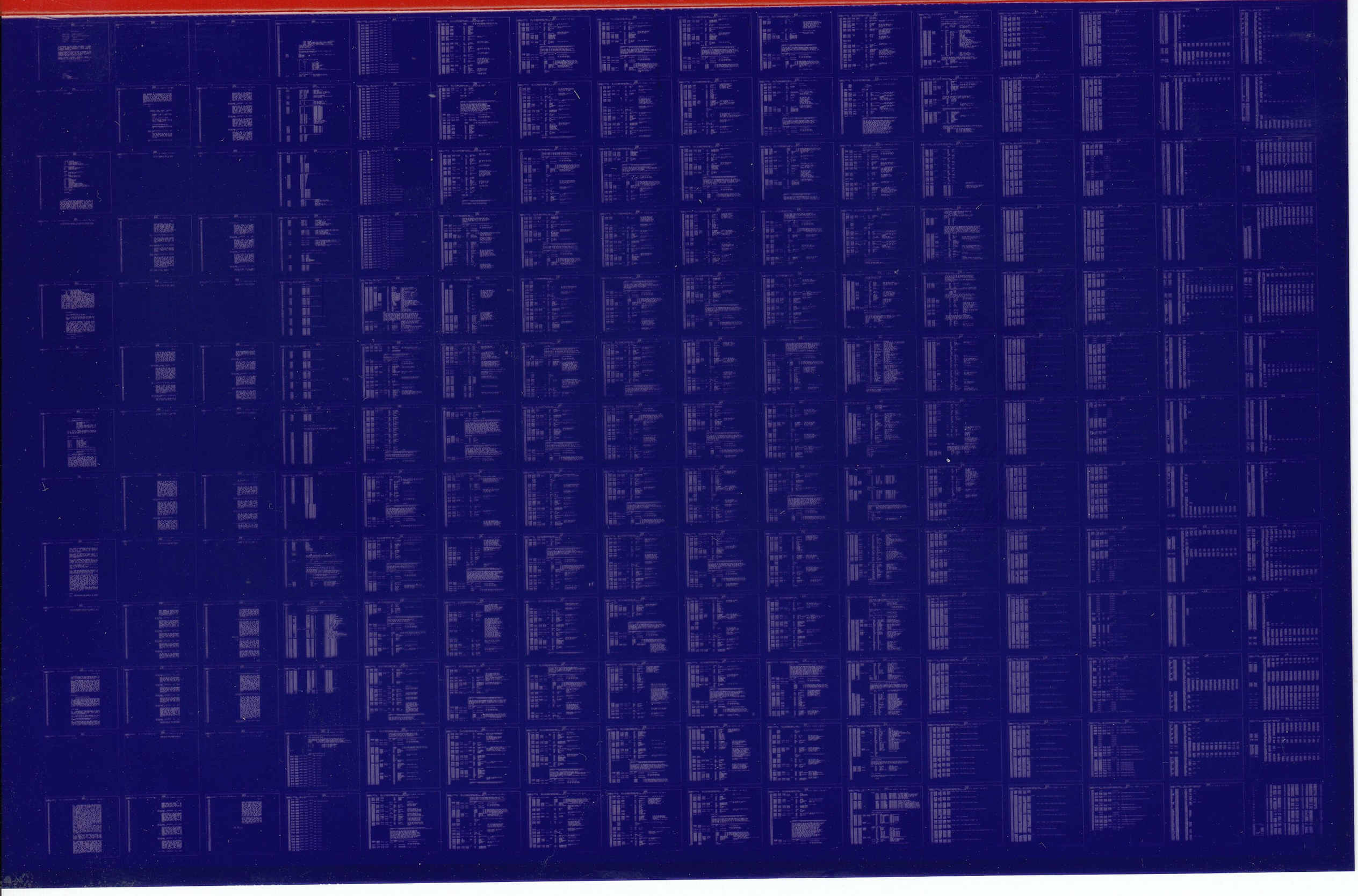
EP DEKBC DL A

NOV 1976

FICHE 1 OF 2

digital

MADE IN USA



PDP-11/70

CACHE DIAG PART 1

AH-0010B-MC

EP-DEKBC-B-DL-A

FICHE 1 OF 2

NOV 1976

digital

MADE IN USA

PDP11/70

CACHE DIAGNOSTIC PART 1

MD-11-DEKBC-B

EP-DEKBC-DL-A

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA

PDP-11/70

CACHE DIAG PART 1
AH-0010B-MC

EP-DEKBC-B-DL-A

FICHE 2 OF 2

NOV 1976
digital
MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DEKBC-B-D

PRODUCT NAME: PDP-11/70 CACHE DIAGNOSTIC PART 1

DATE CREATED: 11-SEPT-75

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: ANTHONY VEZZA

CONTENTS

```
57
58                          3.      LOADING PROCEDURE
                            3.1     METHOD
```

1. ABSTRACT

THE PROGRAMS, DEKBC AND DEKBD, ARE INTENDED TO BE USED AS AIDS FOR THE REPAIR AND MAINTENANCE OF THE CACHE MEMORY SYSTEM IN THE PDP 11/70 COMPUTING SYSTEM. THE AIM IS TO DETECT AND REPORT FAILING COMPONENTS OF THE CACHE UNIT. THE FAILURES ARE TYPICALLY IDENTIFIED WITH A FAILING CIRCUIT WHEN THE REPORT IS MADE, BUT THE OVERALL DIAGNOSTIC PHILOSOPHY HAS BEEN TO LOCATE THE FAILING MODULE (HEX BOARD) OF WHICH THERE ARE FOUR (4) IN THE CACHE UNIT. NOTE THAT WHEN IS FAILURE IS REPORTED AND THE ASSOCIATED CIRCUIT IDENTIFIED, THAT CIRCUIT SHOULD NOT BE TAKEN IN BLIND FAITH

```
                        AS THE DEFECTIVE COMPONENT;  THE IDENTIFIED COMPONENT SHOULD
                        RATHER BE TAKEN AS THE PROBABLE CAUSE OF THE FAILURE.  THERE
```

ARE FOUR (4) MODULES (HEX BOARDS) IN THE CACHE UNIT:

CCB CACHE CONTROL BOARD
CDP CACHE DATA PATHS BOARD
ADM CACHE ADDRESS MEMORY BOARD
DTM CACHE DATA MEMORY BOARD THE PROGRAM, DEKBC, IS DESIGNED TO TEST THE FIRST TWO OF THESE BOARDS; THE PROGRAM, DEKBD, IS DESIGNED TO TEST THE LAST TWO BOARDS. NOTE THAT THOUGH THE TESTING HAS BEEN DIVIDED INTO TWO STAND ALONE PROGRAMS EACH ASSOCIATED WITH TWO MODULES IT SHOULD NOT BE ASSUMED THAT A PARTICULAR MODULE IS WORKING AFTER HAVING RUN ONLY ONE OF THE PROGRAMS! BOTH PROGRAMS SHOULD BE RUN! FOR EXAMPLE, JUST RUNNING DEKBC WITHOUT ERROR DOES NOT RULE OUT A FAILTY COMPONENT ON THE CCB (CACHE CONTROL) BOARD. TO PUT IT MORE SIMPLY THE TESTING HAS BEEN DIVIDED INTO TO PROGRAMS ONLY BECAUSE OF THE RESTRICTIONS OF CORE SIZE! AND NOT TO PROVIDE A MEANS OF TESTING TWO OF THE BOARDS WITH ONE PROGRAM AND THE OTHER TWO BOARDS WITH A SECOND PROGRAM. NOTE THAT DEKBD IS DESIGNED TO RUN AFTER DEKBC. IF THIS HIERARCHY IS NOT HEEDED, THAT IS IF DEKBD IS RUN BEFORE DEKBC, THEN THE ERROR REPORTING FROM DEKBD SHOULD NOT BE STRICTLY INTERPRETED.

2. REQUIREMENTS

2.1 EQUIPMENT PDP 11/70 CPU WITH OPERATORS CONSOLE LA30 OR EQUIVALENT TERMINAL.

2.2 STORAGE BOTH PROGRAMS, DEKBC AND DEKBD, EACH REQUIRE 13K TO LOAD, BUT THEY BOTH ALSO ASSUME THAT THERE IS A MINIMUM OF 28K OF MEMORY IN WHICH TO RUN TESTS.

2.3 PRELIMINARY PROGRAMS THIS PROGRAM ASSUME THAT THE CPU IS FUNCTIONAL! THIS COULD IN SOME CIRCUMSTANCES MEAN THAT THE CPU DIAGNOSTICS SHOULD BE RUN BEFORE EITHER OF THESE DIAGNOSTICS. BUT A FAULTY MEMORY SYSTEM MAY PRECLUDE THIS, SO SITUATIONAL JUDGEMENT MUST BE USED. IF THE CPU IS KNOWN TO BE WORKING THEN RUN THESE DIAGNOSTICS, DEKBC AND DEKBD, FIRST. BUT IF THE CPU CAN NOT BE ASSUMED TO BE WORKING THEN TRY TO RUN THE CPU DIAGNOSTICS FIRST. THEN RUN THESE PROGRAMS IN THE ORDER: DEKBC BEFORE DEKBD! IN FACT DEKBD ASSUMES THAT MUCH OF WHAT IS TESTED IN DEKBC IS OPERATIONAL FOR DOING ITS FAULT ANALYSIS.

3. LOADING PROCEDURE

3.1 METHOD (TO BE SUPPLIED)

4. STARTING PROCEDURE

```
173
174                          4.1     CONTROL SWITCH SETTINGS (SEE 5.1)
```

4.2 STARTING ADDRESS 200

4.3 PROGRAM AND OPERATOR ACTION BOTH PROGRAMS CAN BE STARTED BY:

1 LOAD PROGRAM INTO MEMORY
2 LOAD ADDRESS 200
3 PRESS START
4 THE PROGRAMS WILL LOOP UNTIL THE HALT SWITCH IS PRESSED OR UNTIL THE USER STRIKES (TYPES) CONTROL-C (↑C) ON THE TELETYPE OR TERMINAL (SEE 8.6 AND 5.2.7).

4.4 SPECIAL OPERATOR INTERVENTION OPTIONS IF SWITCH 7 OF THE SWITCH REGISTER IS ON THEN DEKBD WILL REQUIRE THE OPERATOR TO POWER THE MACHINE FIRST DOWN AND THEN UP (SEE 5.1 AND 8.7).

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS FOR DEKBC:

SW<15>=1	HALT ON ERROR
SW<14>=1	LOOP ON TEST
SW<13>=1	INHIBIT ERROR TYPOUTS
SW<12>	NOT USED IN DEKBC
SW<11>=1	INHIBIT ITERATIONS
SW<10>=1	RING BELL ON ERROR
SW<9> =1	LOOP ON ERROR
SW<8> =1	LOOP ON TEST IN SW<7:0>
SW<7:0>	TEST NUMBER FOR LOOPING WHEN SW<8>=1

DEKBD USES THE SAME SWITCH SETTINGS AS DEKBC EXCEPT

SW<7> =1	RUN THE OPERATOR INTERVENTION NEEDED POWER UP TEST

5.2 SUBROUTINE ABSTRACTS BOTH DEKBC AND DEKBD USE THE FOLLOWING SUBROUTINES.

5.2.1 SPURIOUS ERROR HANDLERS THESE ARE TWO ROUTINES WHICH ARE CALLED BY UNEXPECTED TRAPS TO EITHER VECTOR 4, IN THE CASE OF A CPU ERROR, OR VECTOR 114, IN CASE OF A MEMORY PARITY ERROR. THE CPU ERROR HANDLER, CPSPUR, TYPES OUT THE PC AT THE TIME OF THE TRAP AND THE CONTENTS OF THE CPU ERROR REGISTER, CPUERR AND SKIPS TO THE TEST FOLLOWING THE ONE DURING WHICH THE ERROR OCCURRED. THE PARITY ERROR HANDLER, SPUR, TYPES OUT THE PC AT THE TIME OF THE TRAP AND THE CACHE ERROR REGISTERS, MEMERR AND LOADRS AND HIADRS, IT THEN ALSO GIVES CONTROL TO THE TEST FOLLOWING THE ONE DURING WHICH THE ERROR

231 OCCURRED.

5.2.2 SCOPE THIS SUBROUTINE IS CALLED (VIA AN IOT INSTRUCTION) AT THE BEGINNING OF THE EXECUTION OF ALL THE TESTS. IT CONTROLS THE OPERATIONAL FUNCTIONS OF LOOPING ON TEST, ITERATION, AND SETS UP FOR LOOPING ON ERRORS.

5.2.3 ERROR THIS SUBROUTINE IS CALLED (VIA AN EMT INSTRUCTION) TO TYPE OUT AN ERROR REPORT. IT CONTROLS THE OPERATIONAL FUNCTIONS OF HALTING ON ERROR, INHIBITING ERROR PRINT OUT, LOOPING ON ERROR, BELL ON ERROR, ETC.

5.2.4 TRAP CATCHER THIS CONSISTS OF A '.+2' FOLLOWED BY A HALT INSTUCTION REPEATED FROM LOCATION 0 THROUGH 776 FOR THE PURPOSE OF CATCHING ANY SPURIOUS TRAP TO A VECTOR. SUCH A TRAP WILL RESULT IN A HALT AT THE TRAP VECTOR ADDRESS PLUS TWO (2).

5.2.5 TRAP A NUMBER OF SUBROUTINES ARE CALLED BY USING THE TRAP INSTRUCTION:
TYPE TO TYPE OUT AN ASCIZ STRING
TYPEOC TO TYPE OUT THE OCTAL FOR A 16-BIT BINARY NUMBER ETC.

5.2.6 POWER DOWN AND POWER UP THIS SUBROUTINE IS CALLED WHEN AN UNEXPECTED POWER DOWN OCCURS. WHEN POWER IS RETURNED (IF THE HALT SWITCH IS NOT ON) THE PROGRAM WILL RESTART AFTER TYPING A MESSAGE.

5.2.7 MONITOR OR LOADER RESTORE WHEN THIS PROGRAM IS FIRST STARTED IT SAVES THE CONTENTS OF THE HIGHEST 1.5 (DEC) K OF MEMORY IN THE FIRST 28K. THESE LOCATIONS USUALLY CONTAIN THE LOADER OR MONITOR OF THE SYSTEM. TO RESTORE THIS LOADER OR MONITOR THE USER NEED ONLY TYPE CONTROL C (↑C) ON THE TERMINAL AND THAT MONITOR OR LOADER WILL AUTOMATICALLY BE RESTORED. AFTER THIS IS DONE THE PROGRAM WILL HALT. NOTE THAT MANY OF THESE TESTS WIPE OUT THE ORIGINAL CONTENTS OF THAT PART OF MEMORY THEREFORE THE USER SHOULD TYPE CONTROL-C (↑C) TO RESTORE THESE LOCATIONS AND AVOID HAVING TO RELOAD HIS MONITOR OR LOADER.

5.3 OPERATOR ACTION ONLY THE POWER UP INVALIDATOR TEST IN PROGRAM DEKBD REQUIRES OPERATOR INTERVENTION, IN THE FORM OF POWERING THE PROCESSOR FIRST DOWN AND THEN UP. THIS TEST IS RUN ONLY IF SW<12>=1 (SEE 4.4 AND 5.1).

6. ERRORS

6.1 ERROR HALTS ONLY TEST NUMBER 14 IN PROGRAM DEKBC, THE MAINTENANCE REGISTER COUNT PATTERN TEST,

```
                        HALTS THE PROCESSOR IN THE SITUATION WHERE IT  CAN'T
                        CLEAR THE MAINTENANCE REGISTER.  HERE PROCEDING WITH
```

THE PROGRAM'S EXECUTION WOULD PROBABLY BE FATAL, SO A HALT IS EXECUTED! NO OTHER TEST IN EITHER PROGRAM SHOULD HALT UNDER ANY NORMAL ERROR DETECTION.

6.2 ERROR RECOVERY IF NONE OF THE ERROR PERTAINENT OPERATIONAL SWITCHES ARE BEING USED THE PROGRAM WILL EITHER RESUME THE TEST THAT MADE THE ERROR CALL OR START EXECUTION OF THE TEST FOLLOWING THE TEST DURING WHICH THE ERROR CALL WAS MADE DEPENDING ON WHETHER OR NOT THE ERROR WHICH WAS DETECTED (OR EVEN THE ERROR CALL ITSELF) WAS FATAL TO THE TEST WHICH MADE THE ERROR CALL. IF THE HALT DESCRIBED IN 6.1 ABOVE IS EVER EXECUTED TO USER CAN RESUME, IF HE IS BRAVE, BY HITTING THE CONSOLE CONTINUE SWITCH. IF ANY OF THE PERTAINENT CONSOLE SWITCH SETTING ARE SET SEE SECTION 5.1 FOR A DESCRIPTION OF THE ACTION TAKEN WHEN AN ERROR CALL IS MADE.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS NONE

7.2 OPERATING RESTRICTIONS THE MONITOR OR LOADER (OR WHAT EVER IS IN THE FIRST 28K OF MEMORY FROM LOCATIONS 152000 THROUGH LOCATION 157776 ARE SAVED SO THAT THE USER CAN RESTORE HIS LOADER OR MONITOR BY TYPING CONTROL-C (↑C) (SEE 4.3 AND 5.2.7). IF THE PROGRAM WAS CHAINED IN BY A MONITOR WHICH WANTS CONTROL AUTOMATICALLY PASSED BACK TO IT WHEN TESTING IS DONE THAT MONITOR IS RESTORED AND CONTROL IS GIVEN TO IT BY THE END OF PASS ROUTINE .$EOP.

8. MISCELLANEOUS

8.1 EXECUTION TIME FIRST PASS UNDER 10 SECONDS FOR BOTH PROGRAMS. SUBSEQUENT PASSES UNDER 2 MINUTES FOR BOTH PROGRAMS. (MORE EXACT EXECUTION TIMES WILL BE LATER SUPPLIED).

8.2 STACK POINTER IN BOTH PROGRAMS THE STACK POINTER (R6) WILL BE INITIALIZED TO LOCATIO 1100.

8.3 PASS COUNT BOTH PROGRAMS WILL TYPE OUT THE PASS COUNT AT THE END OF EACH PASS.

8.4 ITERATIONS EACH TEST HAS BEEN ASSIGNED AN ITERATION COUNT WHICH WILL DESIGNATE HOW MANY TIMES THAT TEST IS TO BE EXECUTED ON EACH PASS. NOTE THAT ON THE FIRST PASS THE ITERATION COUNT IS OVERIDED BY A ONE (1) MAKING ITERATIONS MEANINGLESS ON THAT

346 FIRST PASS.

8.5 OSCILLOSCOPE SYNC POINTS WHERE EVER POSSIBLE EACH TEST HAS BEEN GIVEN AN OSCILLOSCOPE SYNC POINT (A NOP INSTRUCTION). THE ADDRESS OF THE CONDITICN CODE ROM STATE (44) IS PUT IN THE PROCESSOR MICROBREAK REGISTER (177770). THIS WILL RESULT IN PIN AE1 (SLOT 10) ON THE BACK PLANE TO GO HIGH WHENEVER THE CPU ROM FLOW GOES THROUGH THE MICRO CODE ADDRESS 144. THEREFORE BY USING THE OUTPUT OF THIS BACKPLANE PIN AS A SCOPE SYNC AND BY PUTTING NOP INSTRUCTION IN CRUCIAL PARTS OF A TEST THE USER WILL HAVE A VERY CONVENIENT SYNC FOR MANY SIGNALS HE MAY WISH TO OBSERVER. THE LIMITATIONS OF THIS PROCEDURE ARE THAT THE USER MUST BE ABLE TO JUDGE (DETERMINE) HOW SOON AFTER THE NOP IN THE PARTICULAR TEST HE IS RUNNING (LOOPING ON) THE SIGNAL HE WISHES TO OBSERVE SHOULD OCCUR. IN MANY CASES THIS WILL BE EASY (E.G. THE ERROR REGISTER TESTS.) BUT IN SOME TESTS THE NOP IS SO FAR FROM THE EXPECTED OCCURRENCE OF THE DESIRED SIGNAL THAT THE PROBLEM BECOMES NONTRIVIAL AND THE EXPERIENCED USER WOULD DO WELL TO FIND OTHER SYNC SIGNALS ORIGINATING IN THE CACHE DEVICE ITSELF TO OBSERVE THE LOGIC.

8.6 RESTORING THE MONITOR OR LOADER FOR THE USERS CONVENIENCE BOTH PROGRAMS SAVE EITHER THE MONITOR OR LOADER (OR WHATEVER IS IN THE HIGHEST 1.5K OF MEMORY'S FIRST 28K) AND RESTORE IT WHEN THE USER TYPES CONTROL-C (↑C) ON THE TELETYPE OR TERMINAL. THE PROGRAM WHEN IT GETS THE CONTROL-C RESTORES THE MONITOR AND THEN HALTS; AT THIS POINT THE USERS CAN EITHER RESTART THE MONITOR OR REUSE THE LOADER ETC.

8.7 POWER UP LOGIC TEST THERE IS A CERTAIN PART OF THE CACHE DEVICE WHICH REQUIRES A POWER DOWN POWER UP SEQUENCE TO TEST. THIS TEST HAS BEEN INCLUDED HERE AS AN OPTION ONLY BECAUSE IT REQUIRES OPERATOR INTERVENTION. TO RUN THIS TEST SET SW<12>=1 (SEE 5.1).

8.8 MEMORY MANAGEMENT RESTRICTION OPTION MANY OF THE TESTS REQUIRE THE USE OF EXTENSIVE MEMORY MANAGEMENT MAPPING FACILITY. THESE TESTS MUST ASSUME THE MEMORY MANAGEMENT (AND SOME THE MAPPING BOX) IS OPERATIONAL. NORMALLY THESE TEST WILL BE EXECUTED. BUT THE FEATURE HAS BEEN PROVIDED WHEREBY THE USER CAN DELETE THE EXECUTION OF ANY TESTS WHICH REQUIRE THE USE OF MEMORY MANAGEMENT AND/OR THE MAPPING. THIS HAS BEEN IMPLIMENTED USING SW<7>. WHEN THIS SWITCH IS 0 NORMAL OPERATION IS UNDERTAKEN, BUT WHEN SW<7>=1 THEN ANY TEST WHICH MUST TURN ON THE MEMORY MANAGEMENT UNIT (THE MAPPING BOX) WILL NOT BE RUN AND CONTROL WILL BE PASSED TO THE NEXT TEST!

```
403
404                                 8.9     CRITICAL DEPENDENCE OF  SOME  TESTS  ON  THE
```

CACHE REGISTERS AS THE PROGRAMS RUN FLAGS ARE SET WHICH DESIGNATE THE FUNCTIONALITY OF A CACHE REGISTER. IF A TEST DETERMINES THAT A PARTICULAR REGISTER IS NOT FUNCTIONAL IT SETS A FLAG WHICH DESIGNATES TO THE REST OF THE PROGRAM THAT THAT REGISTER DOES NOT WORK PROPERLY. SOME TESTS WHICH RELY ON THE REGISTERS TO BE FUNCTIONAL WILL TEST THESE FLAGS AND IF THEY FIND THEM TO INDICATE THAT A REGISTER THEY NEED IS BAD THEY WILL SKIP TO THE NEXT TEST!

9. PROGRAM DESCRIPTION

9.1 DEKBC

PROGRAM BY ANTHONY S. VEZZA

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC PACKAGE (MAINDEC-11-DZQAC-A3).

TEST 1 CACHE REGISTERS RESPONSE TEST

REFERENCE EACH CACHE REGISTER MAKING SURE SUCH REFERENCES DO NOT TIME OUT.

TEST 2 CACHE REGISTERS DATA PATH, READ ZEROES TEST

THIS TEST CHECKS THE ABILITY OF THE CACHE REGISTER DATA PATHS TO PASS 0'S BY FIRST WRITING THEN READING 0'S AT THE CONTROL AND MAINTENANCE REGISTERS.

TEST 3 CACHE REGISTERS DATA PATH, READ ONES TEST

```
                                THIS TEST PERFORMS A  READ  OF  BOTH
                                THE  HIGH  ORDER AND LOW ORDER ERROR
```

ADDRESS REGISTER. THIS IS DONE TO MAKE SURE THAT THE REGISTERS' DATA PATHS CAN PASS ONES. NOTE THAT THE LOW ORDER ADDRESS REGISTER SHOULD CONTAIN A 177740 AND THE HIGH ORDER REGISTER SHOULD CONTAIN 000003; THIS LEAVES THE DATA PATH LINE'S BITS 2,3 AND 4 UNTESTED FOR THEIR AVAILITY TO PASS ONES. THIS WILL BE CHECKED IN THE COUNT PATTERN TST4.

TEST 4 CACHE CONTROL REGISTER COUNT PATTERN TEST

THIS TEST RUNS A COUNT PATTERN THROUGH THE CACHE CONTROL REGISTER FOR THE PURPOSE OF CHECKING OUT THE DATA RELIABILITY OF BOTH THE REGISTER BITS AND THE DATA PATHS LINES.

TEST 5 CACHE HIT/MISS AND CONTROL REGISTER SIMPLE MISSES TEST

THIS IS A TEST OF THE HIT/MISS REGISTER AND THE CONTRL REGISTER'S ABILITY TO FORCE MISSES. ZEROES ARE FLOATED THROUGH THE HIT/MISS REGISTER.

TEST 6 CACHE HIT/MISS AND CONTROL REGISTER SIMPLE HIT TEST

THIS IS A TEST OF THE HIT/MISS REGISTER AND THE THE FORCE MISS BITS OF THE CONTROL REGISTER. WHAT IS DONE IS TO SEE IF ANY HITS AT ALL ARE POSSIBLE WITH THE CONTROL REGISTER CLEARED. THEN THE SAME IS DONE WITH EACH GROUP DISABLE ONE AT A TIME. BY DISABLED IS MEANT THAT THE FORCE MISS BIT IS SET IN THE CONTROL REGISTER FOR THE DISABLED GROUP AND THE FORCE SELECT BIT IS SET FOR THE OTHER GROUP.

TEST 7 CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 0 TEST

```
519
520                                     THIS  IS  A  TEST  OF  THE  CONTROL
                                        REGISTER FUNCTIONS OF FORCE MISS AND
```

FORCE SELECTION. AN ADDRESS IS MADE A HIT IN GROUP ONE; THEN ANOTHER ADDRESS, WHOSE HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING SELECTION OF GROUP ZERO; THEN SEE IF THE FIRST ADDRESS IS STILL A HIT IN GROUP ONE; FINALLY TURN ON THE FORCE MISS GROUP ZERO BIT AND SEE IF THE SECOND ADDRESS' HIT IN GROUP ZERO CAN BE FORCED TO A MISS.

TEST 10 CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 1 TEST

THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS MADE A HIT IN GROUP ZERO; THEN ANOTHER ADDRESS, WHOSE HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING SELECTION OF GROUP ONE; THEN SEE IF THE FIRST ADDRESS IS STILL A HIT IN GROUP ZERO; FINALLY TURN ON THE FORCE MISS GROUP ONE BIT AND SEE IF THE SECOND ADDRESS' HIT IN GROUP ONE CAN BE FORCED TO A MISS.

TEST 11 CACHE HIT/MISS REGISTER PATTERNS TEST

THIS IS A TEST OF THE HIT/MISS REGISTER WHICH FLOATS DIFFERENT PATTERNS OF HITS AND MISSES THROUGH THAT REGISTER. THIS IS DONE FIRST WITH BOTH GROUPS ENABLE; THEN WITH GROUP ZERO DISABLED THAT IS FORCING SELECTION OF GROUP ONE AND FORCING MISSES TO GROUP ZERO; FINALLY WITH GROUP ONE DISABLED.

TEST 12 CACHE CONTROL AND HIT/MISS REGISTERS EVALUATION ROUTINE

THIS IS NOT A TEST. THIS ROUTINE IS USED TO LOOK AT THE RESULTS OF TST5 THROUGH TST10, WHICH TESTED THE

```
577                                     HIT/MISS REGISTER AND THE CONTROL
578                                     REGISTER.     THOSE     TESTS     HAVE
```

SIGNALLED A BAD REGISTER USING THE FLAGS, CONFL2 AND HIMFL2, REPRESENTING THE CONTROL AND HIT/MISS REGISTERS RESPECTIVELY. IF ONE OF THESE REGISTERS WAS FOUND TO BE BAD THE FLAG SHOULD BE A -1. WHILE A ZERO FLAG INDICATES THAT THOSE TESTS FOUND THAT REGISTER FUNCTIONAL. THIS ROUTINE LOOKS AT THE FLAGS, CONFL2 AND HIMFL2, WHICH ARE CONSIDERED TO BE LOCAL AND TRANSFERS THE INDICATORS THEY CONTAIN TO THE GLOBAL FLAGS, CONFLG AND HIMFLG. THESE GLOBAL FLAGS ARE USED TO DESIGNATE TO THE REST OF THE PROGRAM THE FUNCTIONALITY OR DISFUNCTIONALITY OF THOSE REGISTERS.

TEST 13 CACHE CONTROL LOGIC, 'RANDOM' FLIP FLOP TEST

THIS IS A TEST OF THE 'RANDOM' CONTROL SIGNAL. A TEST IS MADE TO INSURE THAT THE 'RANDOM' FLIP-FLOP IS NOT STUCK AND IS TOGGLED ONCE FOR EVERY 'BUST' CYCLE INITIATED BY THE PROCESSOR. 'BUST' IS BUS START, A SIGNAL PRODUCED BY THE PROCESSOR WHENEVER IT THINKS IT IS ABOUT TO DO A MEMORY CYCLE. THE RANDOM FLIP FLOP IS USED IN THE CACHE TO DETERMINE WHICH GROUP TO WRITE IN THE EVENT OF A READ MISS CYCLE. IF THIS FLIP FLOP IS SET THEN GROUP ZERO IS WRITTEN; IF CLEAR THEN GROUP ONE IS WRITTEN.

TEST 14 CACHE MAINTENANCE REGISTER COUNT PATTERN TEST

THIS TEST RUNS A COUNT PATTERN THROUGH THE MAINTENANCE REGISTER'S BITS 15 TO 4. THIS IS DONE TO INSURE THAT THESE BITS ARE SETABLE AND THAT THE DATA PATH TO THE REGISTERS IS VIABLE. MISSES ARE FORCED TO BOTH GROUPS SO THAT NO CACHE DATA OR ADDRESS MEMORY ERRORS SHOULD OCCUR. ALSO ANY CYCLES DONE TO MAIN MEMORY ARE INSURED, BY PROPER SELECTION OF INSTRUCTIONS, TO

```
                                RETURN  DATA WITH THE PARITY BITS ON
                                SO  AS  TO  NOT  CAUSE  MAIN  MEMORY
```

PARITY ERRORS BY SETTING THE MAIN MEMORY MAINTENANCE FUNCTION WHICH WOULD EFFECTIVELY FORCE THE PARITY BITS READ FROM MAIN MEMORY TO A ONE. SINCE THESE PARITY ARE ALREADY ONES, NO ERRORS SHOULD OCCUR.

TEST 15 CACHE MAINTENANCE AND ERROR REGISTERS TEST 1

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR ON THE MAIN MEMORY ADDRESS AND CONTROL LINES, AND ALSO A TEST OF THE ERROR REGISTER'S ABILITY TO APPROPRIATELY SET TO 104402. THE REFERENCE CAUSING THIS ERROR WILL BE MADE FROM THE CPU DIRECTLY TO THE CACHE.

TEST 16 CACHE MAINTENANCE AND ERROR REGISTERS TEST 2

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE, WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.

TEST 17 CACHE MAINTENANCE AND ERROR REGISTERS TEST 3

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S HIGH BYTE, WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.

TEST 20 CACHE MAINTENANCE AND ERROR REGISTERS TEST 4

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE, WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.

TEST 21 CACHE MAINTENANCE AND ERROR REGISTERS TEST 5

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S HIGH BYTE, WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.

TEST 22 CACHE MAINTENANCE AND ERROR REGISTERS TEST 6

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE, WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.

TEST 23 CACHE MAINTENANCE AND ERROR REGISTERS TEST 7

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE, WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.

TEST 24 CACHE MAINTENANCE AND ERROR REGISTERS TEST 10

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO, FOR THE LOW BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 25 CACHE MAINTENANCE AND ERROR REGISTERS TEST 11

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY

```
749                                     ERROR IN THE CACHE ADDRESS MEMORY OF
750                                     GROUP ZERO, FOR THE HIGH BYTE OF THE
```

ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 26 CACHE MAINTENANCE AND ERROR REGISTERS TEST 12

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE LOW BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 27 CACHE MAINTENANCE AND ERROR REGISTERS TEST 13

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 30 CACHE MAINTENANCE AND ERROR REGISTERS TEST 14

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ZERO, FOR THE LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 31 CACHE MAINTENANCE AND ERROR

807 REGISTERS TEST 15

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ZERO, FOR THE HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 32 CACHE MAINTENANCE AND ERROR REGISTERS TEST 16

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 33 CACHE MAINTENANCE AND ERROR REGISTERS TEST 17

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 34 CACHE MAINTENANCE AND ERROR REGISTERS TEST 20

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO MAKE THAT REFERENCE CAUSE A MAIN MEMORY ADDRESS AND CONTROL LINES PARITY ERROR ON THE MAIN MEMORY BUS.

```
                                TEST 35 CACHE    MAINTENANCE    AND    ERROR
```

REGISTERS TEST 21

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA PARITY ERROR ON THAT REFERENCE WHICH IS TO AN EVEN WORD IN THE PAIR, WHICH IS ALSO THE WANTED WORD.

TEST 36 CACHE MAINTENANCE AND ERROR REGISTERS TEST 22

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA PARITY ERROR ON THAT REFERENCE WHICH IS TO AN ODD WORD IN THE PAIR, WHICH IS ALSO THE WANTED WORD.

TEST 37 CACHE MAINTENANCE AND ERROR REGISTERS TEST 23

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT ADDRESS .

TEST 40 CACHE MAINTENANCE AND ERROR REGISTERS TEST 24

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE

```
                                                WHICH  RELOCATED  THROUGH THE MEMORY
                                                MANAGEMENT UNIT TO  THE  UNIBUS  AND
```

THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT ADDRESS .

TEST 41 CACHE MAINTENANCE AND ERROR REGISTERS TEST 25

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT DATA .

TEST 42 CACHE MAINTENANCE AND ERROR REGISTERS TEST 26

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT DATA .

TEST 43 CACHE ERROR REGISTER UNIBUS TIME OUT TEST

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO COMPREHEND A CPU TO UNIBUS THROUGH THE MAP TO THE CACHE REFERENCE WHICH TIMES OUT IN MAIN MEMORY. MANY SUCH NON-EXISTENT MEMORY LOCATIONS ARE CONVIENTLY GUARENTEED TO EXIST! ALL THE ADDRESSES FROM 17000000 THROUGH 17777776 ARE ADDRESSES WHICH CAN NOT EXIST. HERE ONLY ONE OF THESE

```
980                                             ADDRESSES, 17777776, WILL BE USED TO
981                                             CAUSE  A  TIME  OUT ON THE UNIBUS AN
```

THE CONSEQUENT ABORT TO VECTOR ERRVEC.

TEST 44 CACHE CONTROL REGISTER DISABLE TRAPS
TEST 1

THIS IS A TEST OF THE CONTROL REGISTER'S ABILITY TO DISABLE A TRAP OCCURRING AS THE RESULT OF A MAIN MEMORY DATA PARITY ERROR IN THE UNWANTED WORD OF THE REFERENCED PAIR. THE MAINTENANCE REGISTER IS USED TO FORCE AN ERROR ON THE LOW BYTE OF THE ODD WORD WHEN REFERENCING THE EVEN WORD OF THAT PAIR.

TEST 45 CACHE CONTROL REGISTER DISABLE TRAPS
TEST 2

THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION. IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE ADDRESS MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO FORCE THE ERROR ON THE LOW BYTE OF THE ADDRESS, IN THE ADDRESS MEMORY OF GROUP 0.

TEST 46 CACHE CONTROL REGISTER DISABLE TRAPS
TEST 3

THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION. IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO FORCE THE ERROR ON THE LOW BYTE OF THE , IN THE MEMORY OF GROUP 0.

TEST 47 CACHE ERROR REGISTER LOCK UP TEST 1

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE ERROR ADDRESS'S ABILITY TO LOCK ON THE

```
1038                                    ADDRESS  OF  THE  FIRST  ERROR  IN A
1039                                    SEQUENCE OF ERRORS.   IN  THIS  TEST
```

TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE CACHE DIRECTLY. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE CACHE DIRECTLY.

TEST 50 CACHE ERROR REGISTER LOCK UP TEST 2

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE CACHE DIRECTLY. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.

TEST 51 CACHE ERROR REGISTER LOCK UP TEST 3

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO

THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE CACHE DIRECTLY.

TEST 52 CACHE ERROR REGISTER LOCK UP TEST 4

THIS IS A TEST OF THE ERRCR REGISTER'S ABILITY TO LOCK UP ON THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.

TEST 53 MAIN MEMORY DATA PARITY CHECKERS LOW BYTE TEST

THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS FOR THE LOW BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD. THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1). THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA PARITY CHECKERS WORKS IN SUCH A WAY AS TO EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS ALREADY ONE THEN NO ERROR OCCURS!

TEST 54 MAIN MEMORY DATA PARITY CHECKERS HIGH BYTE TEST

```
1152
1153                                    THIS IS A TEST OF THE TWO MAIN
```

MEMORY DATA PARITY CHECKERS FOR THE HIGH BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD. THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1). THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA

PARITY CHECKERS WORKS IN SUCH A WAY AS TO EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS ALREADY ONE THEN NO ERROR OCCURS!

```
                        %

                        .LIST   ME
                        .NLIST  MD,MC,CND
```

```
                                .ENABL  ABS,AMA
                                .MCALL  .HEADER,.SWRHI,.1170,.SETUP,.$CATCH,.$ACT11,.$CMTAG
                                .MCALL  .$EOP,.$SCOPE,.$ERROR,.$SAVE,.$TYPE,.$TYPOCT
                                .MCALL  .$TYPDEC,.$TRAP,.$POWER,.$DB20
                        .TITLE  MAINDEC-11-DEKBC-B      PDP 11/70 CACHE DIAGNOSTIC PART 1
                        ;*COPYRIGHT (C) SEPT 11, 1975
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ANTHONY S. VEZZA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-A5).
                        ;*
000001                  $TN=1
160000                  $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
167400                  $SWR=167400
000200                  $SWRMK=200


                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<6:0>
                        ;*         7            SKIP EXECUTION OF ALL TESTS THAT USE MEMORY MANAGEMENT


                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100            ;;FIRST ADDRESS OF THE STACK
001100                  KERSTK= STACK           ;;KERNEL STACK
```

```
                        SUPSTK= STACK-200       ;;SUPERVISOR STACK
                        USESTK= STACK-300       ;;USER STACK
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  SWR=    177570          ;;SWITCH REGISTER
177570                  DISPLAY=SWR

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE LINE FEED
000015                  CR=     15              ;;CODE CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R0,R10          ;;GENERAL REGISTER
                        .EQUIV  R1,R11          ;;GENERAL REGISTER
                        .EQUIV  R2,R12          ;;GENERAL REGISTER
                        .EQUIV  R3,R13          ;;GENERAL REGISTER
                        .EQUIV  R4,R14          ;;GENERAL REGISTER
                        .EQUIV  R5,R15          ;;GENERAL REGISTER
                        .EQUIV  R6,SP           ;;STACK POINTER
                        .EQUIV  SP,KSP          ;;KERNEL STACK POINTER
                        .EQUIV  SP,SSP          ;;SUPERVISOR STACK POINTER
                        .EQUIV  SP,USP          ;;USER STACK POINTER
                        .EQUIV  R7,PC           ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
```

```
                    000400          SW08=   400
                    000200          SW07=   200
                    000100          SW06=   100
                    000040          SW05=   40
                    000020          SW04=   20
                    000010          SW03=   10
                    000004          SW02=   4
                    000002          SW01=   2
                    000001          SW00=   1
                                    .EQUIV  SW09,SW9
                                    .EQUIV  SW08,SW8
                                    .EQUIV  SW07,SW7
                                    .EQUIV  SW06,SW6
                                    .EQUIV  SW05,SW5
                                    .EQUIV  SW04,SW4
                                    .EQUIV  SW03,SW3
                                    .EQUIV  SW02,SW2
                                    .EQUIV  SW01,SW1
                                    .EQUIV  SW00,SW0

                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                    100000          BIT15=  100000
                    040000          BIT14=  40000
                    020000          BIT13=  20000
                    010000          BIT12=  10000
                    004000          BIT11=  4000
                    002000          BIT10=  2000
                    001000          BIT09=  1000
                    000400          BIT08=  400
                    000200          BIT07=  200
                    000100          BIT06=  100
                    000040          BIT05=  40
                    000020          BIT04=  20
                    000010          BIT03=  10
                    000004          BIT02=  4
                    000002          BIT01=  2
                    000001          BIT00=  1
                                    .EQUIV  BIT09,BIT9
                                    .EQUIV  BIT08,BIT8
                                    .EQUIV  BIT07,BIT7
                                    .EQUIV  BIT06,BIT6
                                    .EQUIV  BIT05,BIT5
                                    .EQUIV  BIT04,BIT4
                                    .EQUIV  BIT03,BIT3
                                    .EQUIV  BIT02,BIT2
                                    .EQUIV  BIT01,BIT1
                                    .EQUIV  BIT00,BIT0

                                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                    000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                    000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                    000014          TBITVEC=14              ;;"T" BIT
                    000014          TRTVEC= 14              ;;TRACE TRAP
                    000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                    000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                    000024          PWRVEC= 24              ;;POWER FAIL
```

```
                        EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                        TRAPVEC=34              ;;"TRAP" TRAP
                        TKVEC=  60              ;;TTY KEYBOARD VECTOR
                        TPVEC=  64              ;;TTY PRINTER VECTOR
                        CACHVEC=114             ;;CACHE ERROR INTERRUPT VECTOR
                        PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                        MMVEC=  250             ;;MEMORY MANAGEMENT VECTOR

                        .SBTTL  CACHE   REGISTER DEFINITIONS


 177740                 LOADRS = 177740         ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
 177742                 HIADRS = 177742         ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
 177744                 MEMERR = 177744         ;;CACHE ERROR REGISTER
 177746                 CONTRL = 177746         ;;MEMORY CONTROL REGISTER
 177750                 MAINT  = 177750         ;;MEMORY MAINTENENCE REGISTER
 177752                 HITMIS = 177752         ;;HIT MISS REGISTER "1" IMPLIES HIT IN CACHE


                        .SBTTL  CPU REGISTER DEFINITIONS


 177760                 SIZELO = 177760         ;;MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
                                                ;;TO GET TO THE LAST 32 WORDS OF MEMORY
 177762                 SIZEHI = 177762         ;;HIGH SIZE REGISTER, RESERVED FOR FUTURE USE
                                                ;;CURRENTLY ALL ZERO
 177764                 SYSTID = 177764         ;;SYSTEM ID REGISTER
 177766                 CPUERR = 177766         ;;CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
                                                ;;THE TRAP TO ERRVEC (000004)



                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS


                        ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

 177572                 MMR0=   177572
 177574                 MMR1=   177574
 177576                 MMR2=   177576
 172516                 MMR3=   172516
                        .EQUIV  MMR0,SR0
                        .EQUIV  MMR1,SR1
                        .EQUIV  MMR2,SR2
                        .EQUIV  MMR3,SR3

                        ;*USER "I" PAGE DESCRIPTOR REGISTERS

 177600                 UIPDR0= 177600
 177602                 UIPDR1= 177602
 177604                 UIPDR2= 177604
 177606                 UIPDR3= 177606
 177610                 UIPDR4= 177610
 177612                 UIPDR5= 177612
 177614                 UIPDR6= 177614
```

```
                        177616          UIPDR7= 177616
                                        ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        177620          UDPDR0= 177620
                        177622          UDPDR1= 177622
                        177624          UDPDR2= 177624
                        177626          UDPDR3= 177626
                        177630          UDPDR4= 177630
                        177632          UDPDR5= 177632
                        177634          UDPDR6= 177634
                        177636          UDPDR7= 177636
                                        ;*USER "I" PAGE ADDRESS REGISTERS
                        177640          UIPAR0= 177640
                        177642          UIPAR1= 177642
                        177644          UIPAR2= 177644
                        177646          UIPAR3= 177646
                        177650          UIPAR4= 177650
                        177652          UIPAR5= 177652
                        177654          UIPAR6= 177654
                        177656          UIPAR7= 177656
                                        ;*USER "D" PAGE ADDRESS REGISTERS
                        177660          UDPAR0= 177660
                        177662          UDPAR1= 177662
                        177664          UDPAR2= 177664
                        177666          UDPAR3= 177666
                        177670          UDPAR4= 177670
                        177672          UDPAR5= 177672
                        177674          UDPAR6= 177674
                        177676          UDPAR7= 177676
                                        ;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
                        172200          SIPDR0= 172200
                        172202          SIPDR1= 172202
                        172204          SIPDR2= 172204
                        172206          SIPDR3= 172206
                        172210          SIPDR4= 172210
                        172212          SIPDR5= 172212
                        172214          SIPDR6= 172214
                        172216          SIPDR7= 172216
                                        ;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
                        172220          SDPDR0= 172220
                        172222          SDPDR1= 172222
                        172224          SDPDR2= 172224
                        172226          SDPDR3= 172226
                        172230          SDPDR4= 172230
                        172232          SDPDR5= 172232
                        172234          SDPDR6= 172234
                        172236          SDPDR7= 172236
```

```
                                ;*SUPERVISOR "I" PAGE ADDRESS REGISTERS

            172240              SIPAR0= 172240
            172242              SIPAR1= 172242
            172244              SIPAR2= 172244
            172246              SIPAR3= 172246
            172250              SIPAR4= 172250
            172252              SIPAR5= 172252
            172254              SIPAR6= 172254
            172256              SIPAR7= 172256

                                ;*SUPERVISOR "D" PAGE ADDRESS REGISTERS

            172260              SDPAR0= 172260
            172262              SDPAR1= 172262
            172264              SDPAR2= 172264
            172266              SDPAR3= 172266
            172270              SDPAR4= 172270
            172272              SDPAR5= 172272
            172274              SDPAR6= 172274
            172276              SDPAR7= 172276

                                ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

            172300              KIPDR0= 172300
            172302              KIPDR1= 172302
            172304              KIPDR2= 172304
            172306              KIPDR3= 172306
            172310              KIPDR4= 172310
            172312              KIPDR5= 172312
            172314              KIPDR6= 172314
            172316              KIPDR7= 172316

                                ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS

            172320              KDPDR0= 172320
            172322              KDPDR1= 172322
            172324              KDPDR2= 172324
            172326              KDPDR3= 172326
            172330              KDPDR4= 172330
            172332              KDPDR5= 172332
            172334              KDPDR6= 172334
            172336              KDPDR7= 172336

                                ;*KERNEL "I" PAGE ADDRESS REGISTERS

            172340              KIPAR0= 172340
            172342              KIPAR1= 172342
            172344              KIPAR2= 172344
            172346              KIPAR3= 172346
            172350              KIPAR4= 172350
            172352              KIPAR5= 172352
            172354              KIPAR6= 172354
            172356              KIPAR7= 172356
```

```
                                ;*KERNEL "D" PAGE ADDRESS REGISTERS

 172360                         KDPAR0= 172360
 172362                         KDPAR1= 172362
 172364                         KDPAR2= 172364
 172366                         KDPAR3= 172366
 172370                         KDPAR4= 172370
 172372                         KDPAR5= 172372
 172374                         KDPAR6= 172374
 172376                         KDPAR7= 172376



                                .SBTTL  UNIBUS MAP REGISTER DEFINITIONS

                                ;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
                                ;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

 170200                         MAPL00 = 170200
 170202                         MAPH00 = 170202
 170204                         MAPL01 = 170204
 170206                         MAPH01 = 170206
 170210                         MAPL02 = 170210
 170212                         MAPH02 = 170212
 170214                         MAPL03 = 170214
 170216                         MAPH03 = 170216
 170220                         MAPL04 = 170220
 170222                         MAPH04 = 170222
 170224                         MAPL05 = 170224
 170226                         MAPH05 = 170226
 170230                         MAPL06 = 170230
 170232                         MAPH06 = 170232
 170234                         MAPL07 = 170234
 170236                         MAPH07 = 170236
 170240                         MAPL10 = 170240
 170242                         MAPH10 = 170242
 170244                         MAPL11 = 170244
 170246                         MAPH11 = 170246
 170250                         MAPL12 = 170250
 170252                         MAPH12 = 170252
 170254                         MAPL13 = 170254
 170256                         MAPH13 = 170256
 170260                         MAPL14 = 170260
 170262                         MAPH14 = 170262
 170264                         MAPL15 = 170264
 170266                         MAPH15 = 170266
 170270                         MAPL16 = 170270
 170272                         MAPH16 = 170272
 170274                         MAPL17 = 170274
 170276                         MAPH17 = 170276
 170300                         MAPL20 = 170300
 170302                         MAPH20 = 170302
 170304                         MAPL21 = 170304
```

```
1602    170306          MAPH21 = 170306
1603    170310          MAPL22 = 170310
1604    170312          MAPH22 = 170312
1605    170314          MAPL23 = 170314
1606    170316          MAPH23 = 170316
1607    170320          MAPL24 = 170320
1608    170320          MAPH24 = 170320
1609    170324          MAPL25 = 170324
1610    170326          MAPH25 = 170326
1611    170330          MAPL26 = 170330
1612    170332          MAPH26 = 170332
1613    170334          MAPL27 = 170334
1614    170336          MAPH27 = 170336
1615    170340          MAPL30 = 170340
1616    170342          MAPH30 = 170342
1617    170344          MAPL31 = 170344
1618    170346          MAPH31 = 170346
1619    170350          MAPL32 = 170350
1620    170352          MAPH32 = 170352
1621    170354          MAPL33 = 170354
1622    170356          MAPH33 = 170356
1623    170360          MAPL34 = 170360
1624    170362          MAPH34 = 170362
1625    170364          MAPL35 = 170364
1626    170366          MAPH35 = 170366
1627    170370          MAPL36 = 170370
1628    170372          MAPH36 = 170372
1629    170374          MAPL37 = 170374
1630    170376          MAPH37 = 170376
1631                    .EQUIV  MAPL00,MAPL0
1632                    .EQUIV  MAPH00,MAPH0
1633                    .EQUIV  MAPL01,MAPL1
1634                    .EQUIV  MAPH01,MAPH1
1635                    .EQUIV  MAPL02,MAPL2
1636                    .EQUIV  MAPH02,MAPH2
1637                    .EQUIV  MAPL03,MAPL3
1638                    .EQUIV  MAPH03,MAPH3
1639                    .EQUIV  MAPL04,MAPL4
1640                    .EQUIV  MAPH04,MAPH4
1641                    .EQUIV  MAPL05,MAPL5
1642                    .EQUIV  MAPH05,MAPH5
1643                    .EQUIV  MAPL06,MAPL6
1644                    .EQUIV  MAPH06,MAPH6
1645                    .EQUIV  MAPL07,MAPL7
1646                    .EQUIV  MAPH07,MAPH7
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
```

```
                    000011                  TAB=11
                    000044                  S1M0=44
                    000030                  S0M1=30
                    000054                  S1M0M1=54
                    000034                  S0M0M1=34
                    000014                  M1M0=14
                    000014                  M0M1=M1M0
                    140000                  TESTR1=140000
                    142000                  TESTR2=142000
                    144000                  TESTR3=144000

                                            .SBTTL  TRAP CATCHER

                    000000                          .=0
                                            ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                            ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                            ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

                                            .SBTTL  STARTING ADDRESS(ES)
                    000200                          .=200

000200  000137  003010                              JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM

                                            ;;**************************************************************

                                            .SBTTL          ACT11 HOOKS

                                            ;*THE FOLLOWING LOCATIONS ARE SETUP TO BE USED WITH ACT11
                                            ;*
                                            ;*LOCATION 46 WILL CONTAIN THE ADDRESS OF THE LOCICAL
                                            ;*END OF THE PROGRAM.
                                            ;*LOCATION 52 IS USED TO SPECIFY PROGRAM OPERATING REQUIREMENTS
                                            ;*AND/OR RESTRICTIONS. THIS IS ACCOMPLISHED BY SETTING VARIOUS BITS
                                            ;*TO A ONE OR A ZERO. THE BITS USED AND THERE MEANING ARE:
                                            ;*
                                            ;*      BIT 15=1 PROGRAM SHOULD BE POWER FAILED WHILE RUNNING
                                            ;*           =0 NO POWER FAIL DESIRED
                                            ;*
                                            ;*      BIT 14=1 PROGRAM RUN TIME IS MEMORY SIZE DEPENDENT
                                            ;*           =0 RUN TIME IS NOT MEMORY SIZE DEPENDENT
                                            ;*
                                            ;*      BITS 13-0  MUST BE ZERO'S

                    000204                          $SVPC=.                 ;;SAVE LOCATION COUNTER
                    000046                          .=46                    ;;SET LOCATION COUNTER
000046  026334                              .WORD   $ENDAD                  ;;SET LOC.46 TO ADDRESS $ENDAD
                    000052                          .=52                    ;;SET LOCATION COUNTER
000052  000000                              .WORD   0                       ;;SET LOC.52 TO ZERO
                    000204                          .=$SVPC                 ;; RESTORE LOCATION COUNTER
```

```
                                ;;*****************************************************************

                                .SBTTL  COMMON TAGS

                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                    001100              .=1100

001100                          $CMTAG:                 ;;START OF COMMON TAGS
001100  000000                  $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
001102  000                     $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
001103  000                     $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
001104  000000                  $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                  $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP
001110  000000                  $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                  $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
001114  000                     $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
001115  001                     $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                  $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                  $GDADR: .WORD   0       ;;CONTAINS  OF 'GOOD' DATA
001122  000000                  $BDADR: .WORD   0       ;;CONTAINS  OF 'BAD' DATA
001124  000000                  $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
001126  000000                  $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
001130  000000  000000  000000          .WORD   0,0,0   ;;RESERVED--NOT TO BE USED
001136  177560                  $TKS:   177560          ;;TTY KBD STATUS
001140  177562                  $TKB:   177562          ;;TTY KBD BUFFER
001142  177564                  $TPS:   177564          ;;TTY PRINTER STATUS REG.
001144  177566                  $TPB:   177566          ;;TTY PRINTER BUFFER REG.
001146  000                     $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001147  002                     $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001150  012                     $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001151  000                     $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001152  000000                  $REGAD: .WORD   0       ;;CONTAINS THE  FROM
                                                        ;;WHICH  ($REGO) WAS OBTAINED
001154  000000                  $REG0:  .WORD   0       ;;CONTAINS (($REGAD)+0)
001156  000000                  $REG1:  .WORD   0       ;;CONTAINS (($REGAD)+2)
001160  000000                  $REG2:  .WORD   0       ;;CONTAINS (($REGAD)+4)
001162  000000                  $REG3:  .WORD   0       ;;CONTAINS (($REGAD)+6)
001164  000000                  $REG4:  .WORD   0       ;;CONTAINS (($REGAD)+10)
001166  000000                  $REG5:  .WORD   0       ;;CONTAINS (($REGAD)+12)
001170  000000                  $REG6:  .WORD   0       ;;CONTAINS (($REGAD)+14)
001172  000000                  $REG7:  .WORD   0       ;;CONTAINS (($REGAD)+16)
001174  000000                  $REG10: .WORD   0       ;;CONTAINS (($REGAD)+20)
001176  000000                  $REG11: .WORD   0       ;;CONTAINS (($REGAD)+22)
001200  000000                  $REG12: .WORD   0       ;;CONTAINS (($REGAD)+24)
001202  000000                  $REG13: .WORD   0       ;;CONTAINS (($REGAD)+26)
001204  000000                  $REG14: .WORD   0       ;;CONTAINS (($REGAD)+30)
001206  000000                  $REG15: .WORD   0       ;;CONTAINS (($REGAD)+32)
001210  000000                  $REG16: .WORD   0       ;;CONTAINS (($REGAD)+34)
001212  000000                  $REG17: .WORD   0       ;;CONTAINS (($REGAD)+36)
001214  000000                  $REG20: .WORD   0       ;;CONTAINS (($REGAD)+40)
001216  000000                  $REG21: .WORD   0       ;;CONTAINS (($REGAD)+42)
001220  000000                  $REG22: .WORD   0       ;;CONTAINS (($REGAD)+44)
001222  000000                  $REG23: .WORD   0       ;;CONTAINS (($REGAD)+46)
001224  000000                  $TMP0:  .WORD   0       ;;USER DEFINED
```

```
 1764  001226  000000                $TMP1:   .WORD   0                ;;USER DEFINED
 1765  001230  000000                $TMP2:   .WORD   0                ;;USER DEFINED
 1766  001232  000000                $TMP3:   .WORD   0                ;;USER DEFINED
 1767  001234  000000                $TMP4:   .WORD   0                ;;USER DEFINED
 1768  001236  000000                $TMP5:   .WORD   0                ;;USER DEFINED
 1769  001240  000000                $TMP6:   .WORD   0                ;;USER DEFINED
 1770  001242  000000                $TMP7:   .WORD   0                ;;USER DEFINED
 1771  001244  000000                $TMP10:  .WORD   0                ;;USER DEFINED
 1772  001246  000000                $TMP11:  .WORD   0                ;;USER DEFINED
 1773  001250  000000                $TMP12:  .WORD   0                ;;USER DEFINED
 1774  001252  000000                $TMP13:  .WORD   0                ;;USER DEFINED
 1775  001254  000000                $TMP14:  .WORD   0                ;;USER DEFINED
 1776  001256  000000                $TMP15:  .WORD   0                ;;USER DEFINED
 1777  001260  000000                $TMP16:  .WORD   0                ;;USER DEFINED
 1778  001262  000000                $TMP17:  .WORD   0                ;;USER DEFINED
 1779  001264  000000                $TMP20:  .WORD   0                ;;USER DEFINED
 1780  001266  000000                $TMP21:  .WORD   0                ;;USER DEFINED
 1781  001270  000000                $TMP22:  .WORD   0                ;;USER DEFINED
 1782  001272  000000                $TMP23:  .WORD   0                ;;USER DEFINED
 1783  001274  000000                $TIMES:  0                        ;;MAX. NUMBER OF ITERATIONS
 1784  001276  000000                $ESCAPE: 0                        ;;ESCAPE ON ERROR
 1785  001300  177607  000377        $BELL:   .ASCIZ  <207><377><377>  ;;CODE FOR BELL
 1786  001304     077                $QUES:   .ASCII  /?/              ;;QUESTION MARK
 1787  001305     015                $CRLF:   .ASCII  <15>             ;;CARRIAGE RETURN
 1788  001306  000012                $LF:     .ASCIZ  <12>             ;;LINE FEED
```

```
                                ;;**********************************************************************

                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001310                          $ERRTB:


                                ;ERROR TABLE FOR ERROR TYPE OUT:
                                ;ITEM 1
001310  034763  046334  050460           .WORD   EM1,DH1,DT1,DF1
001316  050263
                                ;ITEM 0
001320  000000  000000  000000           .WORD   0,0,0,0
001326  000000
                                ;ITEM 0
001330  000000  000000  000000           .WORD   0,0,0,0
001336  000000
                                ;ITEM 0
001340  000000  000000  000000           .WORD   0,0,0,0
001346  000000
                                ;ITEM 0
001350  000000  000000  000000           .WORD   0,0,0,0
001356  000000
                                ;ITEM 0
001360  000000  000000  000000           .WORD   0,0,0,0
001366  000000
                                ;ITEM 0
001370  000000  000000  000000           .WORD   0,0,0,0
001376  000000
                                ;ITEM 0
001400  000000  000000  000000           .WORD   0,0,0,0
001406  000000
                                ;ITEM 0
001410  000000  000000  000000           .WORD   0,0,0,0
001416  000000
                                ;ITEM 0
001420  000000  000000  000000           .WORD   0,0,0,0
001426  000000
                                ;ITEM 0
001430  000000  000000  000000           .WORD   0,0,0,0
001436  000000
                                ;ITEM 14
001440  035050  046407  050472           .WORD   EM14,DH14,DT14,DF14
```

```
1845  001446  050267
1846                                    ;ITEM 15
1847  001450  035107  046502  050506              .WORD   EM15,DH15,DT15,DF15
1848  001456  050274
1849                                    ;ITEM 0
1850  001460  000000  000000  000000              .WORD   0,0,0,0
1851  001466  000000
1852                                    ;ITEM 0
1853  001470  000000  000000  000000              .WORD   0,0,0,0
1854  001476  000000
1855                                    ;ITEM 0
1856  001500  000000  000000  000000              .WORD   0,0,0,0
1857  001506  000000
1858                                    ;ITEM 0
1859  001510  000000  000000  000000              .WORD   0,0,0,0
1860  001516  000000
1861                                    ;ITEM 0
1862  001520  000000  000000  000000              .WORD   0,0,0,0
1863  001526  000000
1864                                    ;ITEM 0
1865  001530  000000  000000  000000              .WORD   0,0,0,0
1866  001536  000000
1867                                    ;ITEM 0
1868  001540  000000  000000  000000              .WORD   0,0,0,0
1869  001546  000000
1870                                    ;ITEM 0
1871  001550  000000  000000  000000              .WORD   0,0,0,0
1872  001556  000000
1873                                    ;ITEM 0
1874  001560  000000  000000  000000              .WORD   0,0,0,0
1875  001566  000000
1876                                    ;ITEM 0
1877  001570  000000  000000  000000              .WORD   0,0,0,0
1878  001576  000000
1879                                    ;ITEM 0
1880  001600  000000  000000  000000              .WORD   0,0,0,0
1881  001606  000000
1882
1883                                    ;ITEM 0
1884  001610  000000  000000  000000              .WORD   0,0,0,0
1885  001616  000000
1886                                    ;ITEM 0
1887  001620  000000  000000  000000              .WORD   0,0,0,0
1888  001626  000000
1889                                    ;ITEM 0
1890  001630  000000  000000  000000              .WORD   0,0,0,0
1891  001636  000000
1892                                    ;ITEM 0
1893  001640  000000  000000  000000              .WORD   0,0,0,0
1894  001646  000000
1895                                    ;ITEM 0
1896  001650  000000  000000  000000              .WORD   0,0,0,0
1897  001656  000000
1898                                    ;ITEM 0
1899  001660  000000  000000  000000              .WORD   0,0,0,0
1900  001666  000000
```

```
1901
1902  001670  000000  000000  000000          ;ITEM 0
1903  001676  000000                  .WORD   0,0,0,0
1904
1905  001700  000000  000000  000000          ;ITEM 0
1906  001706  000000                  .WORD   0,0,0,0
1907
1908  001710  000000  000000  000000          ;ITEM 0
1909  001716  000000                  .WORD   0,0,0,0
1910
1911  001720  000000  000000  000000          ;ITEM 0
1912  001726  000000                  .WORD   0,0,0,0
1913
1914  001730  000000  000000  000000          ;ITEM 0
1915  001736  000000                  .WORD   0,0,0,0
1916
1917  001740  000000  000000  000000          ;ITEM 0
1918  001746  000000                  .WORD   0,0,0,0
1919
1920  001750  000000  000000  000000          ;ITEM 0
1921  001756  000000                  .WORD   0,0,0,0
1922
1923  001760  000000  000000  000000          ;ITEM 0
1924  001766  000000                  .WORD   0,0,0,0
1925
1926  001770  000000  000000  000000          ;ITEM 0
1927  001776  000000                  .WORD   0,0,0,0
1928
1929  002000  000000  000000  000000          ;ITEM 0
1930  002006  000000                  .WORD   0,0,0,0
1931
1932  002010  000000  000000  000000          ;ITEM 0
1933  002016  000000                  .WORD   0,0,0,0
1934
1935  002020  000000  000000  000000          ;ITEM 0
1936  002026  000000                  .WORD   0,0,0,0
1937
1938  002030  000000  000000  000000          ;ITEM 0
1939  002036  000000                  .WORD   0,0,0,0
1940
1941  002040  000000  000000  000000          ;ITEM 0
1942  002046  000000                  .WORD   0,0,0,0
1943
1944
1945  002050  035157  046526  050514          ;ITEM 55
1946  002056  050276                  .WORD   EM55,DH55,DT55,DF55
1947
1948  002060  035323  046526  050514          ;ITEM 56
1949  002066  050276                  .WORD   EM56,DH56,DT56,DF56
1950
1951  002070  035470  046526  050514          ;ITEM 57
1952  002076  050276                  .WORD   EM57,DH57,DT57,DF57
1953
1954  002100  035612  046526  050514          ;ITEM 60
1955  002106  050276                  .WORD   EM60,DH60,DT60,DF60
1956
                                      ;ITEM 61
```

```
 1957  002110  035736  046526  050514          .WORD   EM61,DH61,DT61,DF61
 1958  002116  050276
 1959                          ;ITEM 62
 1960  002120  036066  046526  050514          .WORD   EM62,DH62,DT62,DF62
 1961  002126  050276
 1962                          ;ITEM 63
 1963  002130  036214  046603  050526          .WORD   EM63,DH63,DT63,DF63
 1964  002136  050302
 1965                          ;ITEM 64
 1966  002140  036433  046705  050540          .WORD   EM64,DH64,DT64,DF64
 1967  002146  050302
 1968                          ;ITEM 65
 1969  002150  036630  046760  050550          .WORD   EM65,DH65,DT65,DF65
 1970  002156  050302
 1971                          ;ITEM 66
 1972  002160  037213  047062  050562          .WORD   EM66,DH66,DT66,DF66
 1973  002166  050302
 1974                          ;ITEM 67
 1975  002170  037275  047135  050540          .WORD   EM67,DH67,DT67,DF67
 1976  002176  050302
 1977                          ;ITEM 70
 1978  002200  037512  047135  050540          .WORD   EM70,DH70,DT70,DF70
 1979  002206  050302
 1980                          ;ITEM 71
 1981  002210  037770  047135  050540          .WORD   EM71,DH71,DT71,DF71
 1982  002216  050302
 1983                          ;ITEM 72
 1984  002220  040246  047135  050540          .WORD   EM72,DH72,DT72,DF72
 1985  002226  050302
 1986                          ;ITEM 73
 1987  002230  040470  047135  050540          .WORD   EM73,DH73,DT73,DF73
 1988  002236  050302
 1989                          ;ITEM 74
 1990  002240  040754  047135  050540          .WORD   EM74,DH74,DT74,DF74
 1991  002246  050302
 1992
 1993                          ;ITEM 75
 1994  002250  041240  047232  050576          .WORD   EM75,DH75,DT75,DF75
 1995  002256  050307
 1996                          ;ITEM 76
 1997  002260  041240  047232  050612          .WORD   EM76,DH76,DT76,DF76
 1998  002266  050307
 1999                          ;ITEM 77
 2000  002270  041377  047327  050626          .WORD   EM77,DH77,DT77,DF77
 2001  002276  050314
 2002                          ;ITEM 0
 2003  002300  000000  000000  000000          .WORD   0,0,0,0
 2004  002306  000000
 2005                          ;ITEM 0
 2006  002310  000000  000000  000000          .WORD   0,0,0,0
 2007  002316  000000
 2008                          ;ITEM 0
 2009  002320  000000  000000  000000          .WORD   0,0,0,0
 2010  002326  000000
 2011                          ;ITEM 0
 2012  002330  000000  000000  000000          .WORD   0,0,0,0
```

```
002336  000000
                                        ;ITEM 0
002340  000000  000000  000000                .WORD   0,0,0,0
002346  000000
                                        ;ITEM 0
002350  000000  000000  000000                .WORD   0,0,0,0
002356  000000
                                        ;ITEM 0
002360  000000  000000  000000                .WORD   0,0,0,0
002366  000000
                                        ;ITEM 0
002370  000000  000000  000000                .WORD   0,0,0,0
002376  000000
                                        ;ITEM 0
002400  000000  000000  000000                .WORD   0,0,0,0
002406  000000
                                        ;ITEM 0
002410  000000  000000  000000                .WORD   0,0,0,0
002416  000000
                                        ;ITEM 0
002420  000000  000000  000000                .WORD   0,0,0,0
002426  000000

                                        ;ITEM 0
002430  000000  000000  000000                .WORD   0,0,0,0
002436  000000
                                        ;ITEM 0
002440  000000  000000  000000                .WORD   0,0,0,0
002446  000000
                                        ;ITEM 0
002450  000000  000000  000000                .WORD   0,0,0,0
002456  000000
                                        ;ITEM 0
002460  000000  000000  000000                .WORD   0,0,0,0
002466  000000
                                        ;ITEM 117
002470  041535  047232  050612                .WORD   EM117,DH117,DT117,DF117
002476  050307
                                        ;ITEM 120
002500  041664  047353  050654                .WORD   EM120,DH120,DT120,DF120
002506  050326
                                        ;ITEM 121
002510  042077  047427  050744                .WORD   EM121,DH121,DT121,DF121
002516  050361
                                        ;ITEM 122
002520  042300  047471  050756                .WORD   EM122,DH122,DT122,DF122
002526  050365
                                        ;ITEM 123
002530  042430  047553  050756                .WORD   EM123,DH123,DT123,DF123
002536  050365
                                        ;ITEM 124
002540  042631  046407  050770                .WORD   EM124,DH124,DT124,DF124
002546  050371
                                        ;ITEM 0
002550  000000  000000  000000                .WORD   0,0,0,0
002556  000000
```

```
                                        ;ITEM 0
002560  000000  000000  000000                .WORD   0,0,0,0
002566  000000
                                        ;ITEM 127
002570  043037  047723  051010                .WORD   EM127,DH127,DT127,DF127
002576  050415
                                        ;ITEM 130
002600  043221  047765  051042                .WORD   EM130,DH130,DT130,DF130
002606  050401

                                        ;ITEM 131
002610  043273  050043  051054                .WORD   EM131,DH131,DT131,DF131
002616  050420
                                        ;ITEM 132
002620  045406  047613  051010                .WORD   EM132,DH132,DT132,DF132
002626  050401
                                        ;ITEM 133
002630  045545  047650  051020                .WORD   EM133,DH133,DT133,DF133
002636  050405
                                        ;ITEM 134
002640  045717  050122  051102                .WORD   EM134,DH134,DT134,DF134
002646  050432
                                        ;ITEM 135
002650  046065  047327  051122                .WORD   EM135,DH135,DT135,DF135
002656  050441
                                        ;ITEM 0
002660  000000  000000  000000                .WORD   0,0,0,0
002666  000000
                                        ;ITEM 0
002670  000000  000000  000000                .WORD   0,0,0,0
002676  000000
                                        ;ITEM 140
002700  043520  045324  045374                .WORD   EM140,DH140,DT140,DF140
002706  045367
                                        ;ITEM 141
002710  044061  045324  045374                .WORD   EM141,DH141,DT141,DF141
002716  045367
                                        ;ITEM 142
002720  044421  045324  045374                .WORD   EM142,DH142,DT142,DF142
002726  045367
                                        ;ITEM 143
002730  044763  045324  045374                .WORD   EM143,DH143,DT143,DF143
002736  045367

                                        ;ITEM 0
002740  000000  000000  000000                .WORD   0,0,0,0
002746  000000

                                        ;ITEM 0
002750  000000  000000  000000                .WORD   0,0,0,0
002756  000000

                                        ;ITEM 0
002760  000000  000000  000000                .WORD   0,0,0,0
002766  000000

                                        ;ITEM 0
002770  000000  000000  000000                .WORD   0,0,0,0
002776  000000

                                        ;ITEM 150
```

```
                        046250  050177  051150          .WORD   EM150,DH150,DT150,DF150
                        050453
                        005037  001102          START:  CLR     $TSTNM
                        012737  000340  177776          MOV     #340,@#PS       ;;LOCK OUT ALL INTERRUPTS
                        012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
                        005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
                        022706  001136                  CMP     #$TKS,R6        ;;DONE?
                        001374                          BNE     .-6             ;;LOOP BACK IF NO
                        012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                        012737  026370  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
                        012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
                        012737  026644  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
                        012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
                        012737  030016  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
                        012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
                        012737  030076  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
                        012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
                        013737  026264  026256          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
                        005037  001274                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
                        005037  001276                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
                        112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
                        012737  003146  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                        012737  003154  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                        005227  177777                  INC     #-1             ;;FIRST TIME?
                        001043                          BNE     64$             ;;BRANCH IF NO
                        022737  026334  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
                        001437                          BEQ     64$             ;;BRANCH IF YES
                        104400  003206                  TYPE    ,65$            ;;TYPE ASCIZ STRING
                        000434                          BR      64$             ;;GET OVER THE ASCIZ
                                                ;;65$:  .ASCIZ  <CRLF>'MAINDEC-11-DEKBC-B  PDP 11/70 CACHE DIAGNOSTIC PART 1'<CRLF>
                        003276                  64$:
                                                ;THIS ROUTINE SAVES THE TOP 1500 (DEC) WORDS OF THE FIRST 28K OF
                                                ;MEMORY. THESE LOCATIONS SHOULD CONTAIN EITHER THE MONITOR OR THE
                                                ;LOADER WHICH LOADED THE PROGRAM. NOTE THAT TO RESTORE THIS PART
                                                ;OF CORE, THAT IS TO RESTORE THE LOADER OR MONITOR, ALL THE USER
                                                ;MUST DO IS TYPE ↑C (CONTROL-C), WHILE THIS PROGRAM IS RUNNING.
                                                ;THIS WILL AUTOMATICALLY RESTORE THE TOP PART OF MEMORY TO ITS STATE
                                                ;BEFORE THIS PROGRAM WAS STARTED! AFTER THE MONITOR (OR LOADER) HAS BEEN
                                                ;RESTORED THIS PROGRAM WILL HALT.
                        005237  031130          LOOP:   INC     MONF            ;INCREMENT THE FLAG WHICH INDICATES
                        001013                          BNE     TOP             ;WHETHER OR NOT THE TOP OF MEMORY
                                                                                ;IN THE FIRST 28K HAS BEEN SAVED.
                        013737  000060  031126          MOV     @#TKVEC,MONTTY  ;SAVE THE INITIAL CONTENTS OF THE TTY KEYBOARD
                                                                                ;VECTOR.
                        012700  002734                  MOV     #↑D1500,R0      ;IF NOT THEN SAVE IT.
                        012701  051166                  MOV     #BOTTOM+4,R1    ;SAVE IT AT THE BOTTOM OF THIS PROGRAM.
                        012702  160000                  MOV     #160000,R2      ;GET THE ADDRESS OF THE END OF THE MONITOR.
                        014221                  1$:     MOV     -(R2),(R1)+     ;SAVE 1500 (DEC) LOCATIONS (WORDS)
                        077002                          SOB     R0,1$
                        012737  000044  177770  TOP:    MOV     #44,@#177770    ;SET TO SYNC SCOPE (OSCILLOSCOPE)
                                                                                ;ON A NOP INSTRUCTION.
                        012737  031006  000060          MOV     #RESMON,@#TKVEC ;SET UP THE KEYBORD INTERRUPT VECTOR.
```

```
2181  003346  012737  000340  000062          MOV     #340,@#TKVEC+2
2182  003354  005077  175560                  CLR     @$TKB                   ;MAKE SURE THE BUFFER IS CLEAR
2183  003360  152777  000100  175550          BISB    #BIT6,@$TKS             ;TURN ON INTERRUPT ENABLE FOR THE KEYBOARD.
2184
2185  003366  012737  030352  000004          MOV     #CPSPUR,@#ERRVEC        ;SET UP FOR UNEXPECTED ERRORS.
2186  003374  012737  030400  000114          MOV     #SPUR,@#CACHVEC
2187
2188
2189                                          ;;*****************************************************************
2190                                          ;*TEST 1        CACHE REGISTERS RESPONSE TEST
2191                                          ;*
2192                                          ;*REFERENCE EACH CACHE REGISTER MAKING SURE SUCH
2193                                          ;*REFERENCES DO NOT TIME OUT.
2194                                          ;*
2195                                          ;;*****************************************************************
2196  003402  000004                          TST1:   SCOPE
2197  003404  012737  000040  001274          MOV     #40,$TIMES              ;;DO 40 ITERATIONS
2198          000001                  JA=$TN-1
2199                                                                          ;SET THE SKAD REGISTER
2200  003412  012737  003756  030524          MOV     #TST2,SKAD              ;IN CASE THE TEST ABORTS.
2201
2202  003420  113737  001102  001224          MOVB    $TSTNM,$TMPO
2203  003426  012737  030400  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
2204  003434  012701  030734                  MOV     #LOAFLG,R1              ;CLEAR THE REGISTER FLAGS
2205  003440  012700  000014                  MOV     #14,R0
2206  003444  005021                  64$:    CLR     (R1)+
2207  003446  077002                          SOB     R0,64$
2208  003450  013737  000004  003526          MOV     @#ERRVEC,JATMP  ;SAVE THE OLD CONTENTS OF VECTOR ERRVEC.
2209  003456  012737  003530  000004          MOV     #JAERR,@#ERRVEC ;SET UP THE TIME OUT
2210                                                                          ;VECTOR
2211  003464  012700  177740                  MOV     #LOADRS,R0
2212  003470  012737  003476  001110          MOV     #JA1,$LPERR
2213
2214  003476  000240                  JA1:    NOP                             ;FOR SCOPING WITH AN OSCILLOSCOPE!
2215  003500  005710                          TST     (R0)                    ;REFERENCE EACH CACHE REGISTER
2216                                                                          ;MAKING SURE EACH DOESN'T TIME OUT.
2217
2218  003502  062700  000002          JA2:    ADD     #2,R0
2219  003506  020027  177752                  CMP     R0,#HITMIS
2220  003512  101771                          BLOS    JA1
2221
2222  003514  013737  003526  000004  JA3:    MOV     JATMP,@#ERRVEC  ;RESET THE CPU TRAP VECTOR.
2223  003522  000137  003752                  JMP     JADONE
2224
2225  003526  000000                  JATMP:  .WORD   0                       ;SAVE THE OLD CONTENTS OF
2226                                                                          ;VECTOR ERRVEC HERE.
2227
2228  003530  032737  000020  177766  JAERR:  BIT     #20,@#CPUERR
2229  003536  001005                          BNE     JAERR1                  ;MAKE SURE THE ERROR
2230  003540  013737  003526  000004  JAERR0: MOV     JATMP,@#ERRVEC  ;IF NOT RESET VECTOR ERRVEC AND GO TO
2231  003546  000177  174232                  JMP     @ERRVEC                 ;THE ROUTINE WHICH HANDLES CPU ERRORS.
2232  003552  021627  003502          JAERR1: CMP     (SP),#JA2               ;OTHERWISE REPORT THE FACT THAT A CACHE
2233  003556  001370                          BNE     JAERR0                  ;REGISTER REFERENCE TIMED OUT!
2234  003560  012637  001226                  MOV     (SP)+,$TMP1
2235  003564  005726                          TST     (SP)+
2236  003566  010037  001232                  MOV     R0,$TMP3
```

```
2237  003572  012737  000077  001234          MOV     #77,$TMP4
2238  003600  020027  177740                  CMP     R0,#LOADRS
2239  003604  001005                          BNE     JAERR2
2240  003606  012737  177777  030734          MOV     #-1,LOAFLG
2241  003614  104055                  1$:     ERROR   55
2242  003616  000451                          BR      JAERR9
2243
2244  003620  020027  177742          JAERR2: CMP     R0,#HIADRS
2245  003624  001005                          BNE     JAERR3
2246  003626  012737  177777  030736          MOV     #-1,HIAFLG
2247  003634  104056                  1$:     ERROR   56
2248  003636  000441                          BR      JAERR9
2249
2250  003640  020027  177744          JAERR3: CMP     R0,#MEMERR
2251  003644  001005                          BNE     JAERR4
2252  003646  012737  177777  030740          MOV     #-1,MMRFLG
2253  003654  104057                  1$:     ERROR   57
2254  003656  000431                          BR      JAERR9
2255
2256  003660  020027  177746          JAERR4: CMP     R0,#CONTRL
2257  003664  001005                          BNE     JAERR5
2258  003666  012737  177777  030742          MOV     #-1,CONFLG
2259  003674  104060                  1$:     ERROR   60
2260  003676  000421                          BR      JAERR9
2261
2262  003700  020027  177750          JAERR5: CMP     R0,#MAINT
2263  003704  001005                          BNE     JAERR6
2264  003706  012737  177777  030744          MOV     #-1,MANFLG
2265  003714  104061                  1$:     ERROR   61
2266  003716  000411                          BR      JAERR9
2267
2268  003720  020027  177752          JAERR6: CMP     R0,#HITMIS
2269  003724  001005                          BNE     JAERR7
2270  003726  012737  177777  030746          MOV     #-1,HIMFLG
2271  003734  104062                  1$:     ERROR   62
2272  003736  000401                          BR      JAERR9
2273
2274  003740  000000          JAERR7: HALT                    ;???
2275
2276  003742  005037  177766  JAERR9: CLR     @#CPUERR
2277  003746  000137  003502          JMP     JA2
2278
2279  003752  005037  177766  JADONE: CLR     @#CPUERR        ;DONE!
2280
2281                          ;;*****************************************************************
2282                          ;*TEST 2         CACHE REGISTERS DATA PATH, READ ZEROES TEST
2283                          ;*
2284                          ;*THIS TEST CHECKS THE ABILITY OF THE CACHE REGISTER
2285                          ;*DATA PATHS TO PASS 0'S BY FIRST WRITING THEN READING
2286                          ;*0'S AT THE CONTROL AND MAINTENANCE REGISTERS.
2287                          ;*
2288                          ;;*****************************************************************
2289  003756  000004          TST2:   SCOPE
2290           000002          JB=$TN-1
2291                                                  ;SET THE SKAD REGISTER
2292  003760  012737  004112  030524          MOV     #TST3,SKAD      ;IN CASE THE TEST ABORTS.
```

```
2293
2294  003766  113737  001102  001224          MOVB    $TSTNM,$TMPO
2295  003774  012737  030400  000114          MOV     #SPUR,@#CACHVEC
2296
2297  004002  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2298  004004  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2299  004006  012737  004014  001110          MOV     #JB1,$LPERR
2300  004014  005037  177746          JB1:    CLR     @#CONTRL        ;WRITE ZEROES
2301  004020  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
2302  004022  013700  177746                  MOV     @#CONTRL,R0     ;READ.ZEROES
2303  004026  005700                          TST     R0
2304  004030  001430                          BEQ     JBDONE
2305  004032  005037  177750          JB2:    CLR     @#MAINT
2306  004036  013701  177750                  MOV     @#MAINT,R1
2307  004042  005701                          TST     R1
2308  004044  001414                          BEQ     JBERR2
2309
2310  004046                          JBERR1:                         ;BOTH READ ZEROES FAILED.
2311  004046  010037  001230                  MOV     R0,$TMP2
2312  004052  010137  001232                  MOV     R1,$TMP3
2313  004056  104063                  1$:     ERROR   63
2314  004060  012737  177777  030742          MOV     #-1,CONFLG      ;SIGNAL BAD REGISTERS
2315  004066  012737  177777  030744          MOV     #-1,MANFLG
2316  004074  000406                          BR      JBDONE
2317
2318  004076                          JBERR2:                         ;ONLY THE READ OF THE
2319  004076  010037  001230                  MOV     R0,$TMP2        ;CONTROL REGISTER FAILED.
2320  004102  104064                  1$:     ERROR   64
2321  004104  012737  177777  030742          MOV     #-1,CONFLG
2322
2323  004112                          JBDONE:                         ;DONE!!!
2324
2325                                          ;;*****************************************************************
2326                                          ;*TEST 3        CACHE REGISTERS DATA PATH, READ ONES TEST
2327                                          ;*
2328                                          ;*THIS TEST PERFORMS A READ OF BOTH THE HIGH ORDER AND
2329                                          ;*LOW ORDER ERROR ADDRESS REGISTER. THIS IS DONE TO MAKE
2330                                          ;*SURE THAT THE REGISTERS' DATA PATHS CAN PASS ONES. NOTE THAT
2331                                          ;*THE LOW ORDER ADDRESS REGISTER SHOULD CONTAIN A
2332                                          ;*177740 AND THE HIGH ORDER REGISTER SHOULD CONTAIN
2333                                          ;*000003; THIS LEAVES THE DATA PATH LINE'S BITS 2,3 AND 4
2334                                          ;*UNTESTED FOR THEIR AVAILITY TO PASS ONES. THIS WILL
2335                                          ;*BE CHECKED IN THE COUNT PATTERN TST4.
2336                                          ;*
2337                                          ;;*****************************************************************
2338  004112  000004                  TST3:   SCOPE
2339  004114  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2340          000003                  JC=$TN-1
2341                                                                  ;SET THE SKAD REGISTER
2342  004122  012737  004254  030524          MOV     #TST4,SKAD      ;IN CASE THE TEST ABORTS.
2343
2344  004130  113737  001102  001224          MOVB    $TSTNM,$TMPO
2345
2346
2347  004136  104426                          SKPBAD          ;IF THE ERROR ADDRESS REG IS BAD SKIP THIS TEST.
2348  004140  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
```

```
2349  004142  012737  177777  177744          MOV     #-1,@#MEMERR    ;MAKE SURE THE ERROR REGISTERS ARE UNLOCKED
2350  004150  012737  004156  001110          MOV     #JC1,$LPERR
2351
2352  004156  000240                  JC1:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
2353  004160  013700  177740                  MOV     @#LOADRS,R0
2354  004164  013701  177742                  MOV     @#HIADRS,R1     ;READ THE REGISTERS.
2355  004170  022700  177740                  CMP     #177740,R0
2356  004174  001003                          BNE     JCERR1
2357  004176  022701  000003          JC2:    CMP     #3,R1
2358  004202  001424                          BEQ     JCDONE
2359
2360  004204  012737  004222  001226  JCERR1: MOV     #1$,$TMP1       ;BAD DATA WAS READ FROM THEM!!
2361  004212  010037  001230                  MOV     R0,$TMP2
2362  004216  010137  001232                  MOV     R1,$TMP3
2363  004222  104065                  1$:     ERROR   65
2364  004224  022700  000003                  CMP     #3,R0
2365  004230  001403                          BEQ     2$
2366  004232  012737  177777  030734          MOV     #-1,LOAFLG
2367  004240  022700  177740          2$:     CMP     #177740,R0
2368  004244  001403                          BEQ     JCDONE
2369  004246  012737  177777  030736          MOV     #-1,HIAFLG
2370
2371  004254                          JCDONE: ;DONE!
2372
2373
2374                                  ;;**********************************************************************
2375                                  ;*TEST 4         CACHE CONTROL REGISTER COUNT PATTERN TEST
2376                                  ;*
2377                                  ;*THIS TEST RUNS A COUNT PATTERN THROUGH THE CACHE CONTROL
2378                                  ;*REGISTER FOR THE PURPOSE OF CHECKING OUT THE
2379                                  ;*DATA RELIABILITY OF BOTH THE REGISTER BITS AND THE
2380                                  ;*DATA PATHS LINES.
2381                                  ;*
2382                                  ;;**********************************************************************
2383  004254  000004                  TST4:   SCOPE
2384  004256  012737  000004  001274          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
2385
2386          000004                  JD=$TN-1
2387                                                                  ;SET THE SKAD REGISTER
2388  004264  012737  004372  030524          MOV     #TST5,SKAD      ;IN CASE THE TEST ABORTS.
2389
2390  004272  113737  001102  001224          MOVB    $TSTNM,$TMP0
2391
2392
2393  004300  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2394
2395  004302  012700  177746                  MOV     #CONTRL,R0
2396  004306  005002                          CLR     R2
2397  004310  012737  004316  001110          MOV     #JD1,$LPERR
2398  004316  000240                  JD1:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
2399  004320  010210                          MOV     R2,(R0)         ;WRITE THE REGISTER.
2400  004322  011001                          MOV     (R0),R1         ;READ BACK THE REGISTER AND MAKE SURE
2401  004324  010203                          MOV     R2,R3           ;THE DATA IS CORRECT.
2402  004326  042703  177700                  BIC     #177700,R3
2403  004332  020301                          CMP     R3,R1
2404  004334  001003                          BNE     JDERR1
```

```
2405  004336  077211                    JD2:    SOB     R2,JD1
2406  004340  005010                            CLR     (R0)
2407  004342  000413                            BR      JDDONE
2408  004344                            JDERR1:                         ;REPORT THE ERROR!
2409  004344  010237  001230                    MOV     R2,$TMP2
2410  004350  010137  001232                    MOV     R1,$TMP3
2411  004354  010337  001234                    MOV     R3,$TMP4
2412  004360  104066                   1$:     ERROR   66
2413  004362  012737  177777  030742            MOV     #-1,CONFLG
2414  004370  000762                            BR      JD2
2415  004372                            JDDONE:
2416
2417
2418                                    ;;*****************************************************************
2419                                    ;*TEST 5        CACHE HIT/MISS AND CONTROL REGISTER SIMPLE MISSES TEST
2420                                    ;*
2421                                    ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
2422                                    ;*CONTRL REGISTER'S ABILITY TO FORCE MISSES. ZEROES ARE
2423                                    ;*FLOATED THROUGH THE HIT/MISS REGISTER.
2424                                    ;*
2425                                    ;*****************************************************************
2426  004372  000004                    TST5:   SCOPE
2427  004374  012737  000040  001274            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2428           000005                   KB=$TN-1
2429                                                                    ;SET THE SKAD REGISTER
2430  004402  012737  004724  030524            MOV     #TST6,SKAD      ;IN CASE THE TEST ABORTS.
2431
2432  004410  113737  001102  001224            MOVB    $TSTNM,$TMPC
2433
2434
2435  004416  104432                            SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2436  004420  104436                            SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2437  004422  005037  004614                    CLR     KBFLG
2438  004426  012737  000014  177746  KB1:     MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
2439  004434  012737  004426  001110            MOV     #KB1,$LPERR
2440
2441  004442  012700  004452                    MOV     #KB2,R0
2442  004446  012701  000020                    MOV     #20,R1
2443  004452  005720                   KB2:     TST     (R0)+
2444  004454  077102                            SOB     R1,KB2
2445  004456  000240                            NOP                     ;GET SIX FORCED MISSES.
2446  004460  000240                            NOP
2447  004462  000240                            NOP
2448  004464  000240                            NOP
2449  004466  013702  177752                    MOV     @#HITMIS,R2     ;SHOULD HAVE REGISTERED
2450  004472  001051                            BNE     KBERR1          ;SIX MISSES.
2451
2452  004474  012737  004474  001110  KB3:     MOV     #KB3,$LPERR
2453  004502  012737  000054  177746            MOV     #S1MOM1,@#CONTRL        ;SELECT GROUP ONE, MISS GROUP
2454  004510  012700  004520                    MOV     #KB4,R0         ;ZERO AND GROUP ONE.
2455  004514  012701  000020                    MOV     #20,R1
2456  004520  005720                   KB4:     TST     (R0)+
2457  004522  077102                            SOB     R1,KB4
2458  004524  000240                            NOP
2459  004526  000240                            NOP
2460  004530  000240                            NOP
```

```
2461  004532  000240                        NOP
2462  004534  013702  177752                MOV     @#HITMIS,R2     ;SHOULD HAVE SIX MISSES.
2463  004540  001035                        BNE     KBERR2
2464
2465  004542  012737  004542  001110  KB5:  MOV     #KB5,$LPERR
2466  004550  012737  000034  177746        MOV     #SOMOM1,@#CONTRL        ;SELECT GROUP 0, MISS GROUP 0
2467  004556  012700  004566                MOV     #KB6,R0         ;AND GROUP 1.
2468  004562  012701  000020                MOV     #20,R1
2469  004566  005720              KB6:      TST     (R0)+
2470  004570  077102                        SOB     R1,KB6
2471  004572  000240                        NOP
2472  004574  000240                        NOP
2473  004576  000240                        NOP
2474  004600  000240                        NOP
2475  004602  013702  177752                MOV     @#HITMIS,R2     ;SHOULD HAVE SIX MISSES.
2476  004606  001021                        BNE     KBERR3
2477  004610  000137  004666                JMP     KBDONE
2478
2479
2480  004614  000000              KBFLG:    .WORD   0               ;ERROR FLAG.
2481
2482  004616                      KBERR1:                           ;GOT HITS WHILE FORCING
2483  004616  010237  001230                MOV     R2,$TMP2        ;MISSES TO BOTH GROUPS.
2484  004622  104072              1$:       ERROR   72
2485  004624  052737  000001  004614        BIS     #BIT0,KBFLG
2486  004632  000720                        BR      KB3
2487  004634                      KBERR2:                           ;GO HITS WHILE FORCING
2488  004634  010237  001230                MOV     R2,$TMP2        ;MISSES TO BOTH GROUPS
2489  004640  104073              1$:       ERROR   73              ;AND SELECTING GROUP 1
2490  004642  052737  000002  004614        BIS     #BIT1,KBFLG
2491  004650  000734                        BR      KB5
2492  004652                      KBERR3:                           ;GO HITS WHILE FORCING
2493  004652  010237  001230                MOV     R2,$TMP2        ;MISSES TO BOTH GROUPS
2494  004656  104074              1$:       ERROR   74              ;AND SELECTING GROUP 0.
2495  004660  052737  000004  004614        BIS     #BIT2,KBFLG
2496
2497  004666  005037  177746      KBDONE:  CLR     @#CONTRL
2498  004672  022737  000007  004614        CMP     #7,KBFLG        ;IF THE TEST DETECTED
2499  004700  001003                        BNE     KBD2            ;HITS FOR ALL OF THE
2500  004702  012737  177777  030762        MOV     #-1,HIMFL2      ;THREE CONDITION USED IN
2501                                                                ;THE CONTROL REGISTER
2502                                                                ;SIGNAL A BAD HIT/MISS
2503                                                                ;REGISTER.
2504  004710  005737  004614      KBD2:    TST     KBFLG           ;IF LESS THEN THREE (BUT
2505  004714  001403                        BEQ     KBD3            ;MORE THAN ZERO) CONTRL
2506  004716  012737  177777  030756        MOV     #-1,CONFL2      ;PATTERNS FAILED SIGNAL
2507                                                                ;A BAD CONTROL REGISTER.
2508  004724                      KBD3:                             ;DONE!
2509
2510                              ;;*****************************************************************
2511                              ;*TEST 6        CACHE HIT/MISS AND CONTROL REGISTER SIMPLE HIT TEST
2512                              ;*
2513                              ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
2514                              ;*THE FORCE MISS BITS OF THE CONTROL REGISTER.
2515                              ;*WHAT IS DONE IS TO SEE IF ANY HITS AT ALL ARE
2516                              ;*POSSIBLE WITH THE CONTROL REGISTER CLEARED.  THEN THE
```

```
                                        ;*SAME IS DONE WITH EACH GROUP DISABLE ONE AT A TIME.
                                        ;*BY DISABLED IS MEANT THAT THE FORCE MISS BIT IS SET
                                        ;*IN THE CONTROL REGISTER FOR THE DISABLED GROUP AND THE
                                        ;*FORCE SELECT BIT IS SET FOR THE OTHER GROUP.
                                        ;*
                                        ;;**************************************************************
004724  000004                TST6:   SCOPE
004726  012737  000040  001274        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000006              KA=$TN-1
                                                            ;SET THE SKAD REGISTER
004734  012737  005274  030524        MOV     #TST7,SKAD      ;IN CASE THE TEST ABORTS.

004742  113737  001102  001224        MOVB    $TSTNM,$TMP0

004750  104432                        SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
004752  104436                        SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
004754  005037  005160                CLR     KAFLG
004760  005037  177746        KA1:    CLR     @#CONTRL        ;BOTH GROUPS ENABLED.
004764  012737  004760  001110        MOV     #KA1,$LPERR
004772  012700  005002                MOV     #KA2,R0
004776  012701  000020                MOV     #20,R1

005002  005720                KA2:    TST     (R0)+           ;SET UP HITS IN BOTH
005004  077102                        SOB     R1,KA2          ;GROUPS
005006  000240                        NOP
005010  000240                        NOP
005012  000240                        NOP
005014  000240                        NOP
005016  013702  177752                MOV     @#HITMIS,R2     ;SHOULD HAVE ALL HITS.
005022  022702  000077                CMP     #77,R2
005026  001055                        BNE     KAERR1

005030  012737  005030  001110  KA3:  MOV     #KA3,$LPERR
005036  012737  000044  177746        MOV     #S1M0,@#CONTRL  ;DISABLE GROUP ZERO.
005044  012700  005054                MOV     #KA4,R0
005050  012701  000020                MOV     #20,R1
005054  005720                KA4:    TST     (R0)+           ;SET UP HITS IN GROUP 1
005056  077102                        SOB     R1,KA4
005060  000240                        NOP
005062  000240                        NOP
005064  000240                        NOP
005066  000240                        NOP
005070  013702  177752                MOV     @#HITMIS,R2     ;SHOULD HAVE ALL HITS.
005074  022702  000077                CMP     #77,R2
005100  001037                        BNE     KAERR2
005102  012737  005102  001110  KA5:  MOV     #KA5,$LPERR
005110  012737  000030  177746        MOV     #S0M1,@#CONTRL  ;DISABLE GROUP ONE.
005116  012700  005126                MOV     #KA6,R0
005122  012701  000020                MOV     #20,R1
005126  005720                KA6:    TST     (R0)+           ;SET UP HITS IN GROUP ZERO.
005130  077102                        SOB     R1,KA6
005132  000240                        NOP
005134  000240                        NOP
005136  000240                        NOP
005140  000240                        NOP
```

```
2573  005142  013702  177752              MOV     @#HITMIS,R2     ;SHOULD HAVE SIX HITS.
2574  005146  022702  000077              CMP     #77,R2
2575  005152  001021                      BNE     KAERR3
2576  005154  000137  005232              JMP     KADONE
2577
2578  005160  000000          KAFLG:      .WORD   0               ;ERROR FLAG.
2579
2580  005162                  KAERR1:                             ;FAILED TO GET HITS
2581  005162  01C237  001230              MOV     R2,$TMP2        ;WITH THE CONTROL
2582  005166  104067          1$:         ERROR   67              ;REGISTER CLEAR!
2583  005170  052737  000001  005160      BIS     #BIT0,KAFLG
2584  005176  000714                      BR      KA3
2585  005200                  KAERR2:                             ;FAILED TO GET HITS
2586  005200  010237  001230              MOV     R2,$TMP2        ;WITH THE CONTROL REGISTER
2587  005204  104070          1$:         ERROR   70              ;SET TO FORCE SELECT GROUP
2588  005206  052737  000002  005160      BIS     #BIT1,KAFLG     ;ONE FORCE MISS GROUP ZERO.
2589  005214  000732                      BR      KA5
2590  005216                  KAERR3:                             ;FAILED TO GET HITS
2591  005216  010237  001230              MOV     R2,$TMP2        ;WITH THE CONTROL REGISER
2592  005222  104071          1$:         ERROR   71              ;SET TO FORCE SELECT GROUP
2593  005224  052737  000004  005160      BIS     #BIT2,KAFLG     ;ZERO AND FORCE MISS GROUP ONE.
2594  005232  005037  177746  KADONE:     CLR     @#CONTRL
2595  005236  022737  000007  005160      CMP     #7,KAFLG        ;IF THE TEST FAILED FOR ALL
2596  005244  001004                      BNE     KAD2            ;THREE CONDITIONS OF THE
2597  005246  012737  177777  030746      MOV     #-1,HIMFLG      ;CONTROL REGISTER SIGNAL
2598  005254  000407                      BR      KAD3            ;A BAD HIT/MISS REGISTER.
2599
2600  005256  032737  000006  005160  KAD2:   BIT     #6,KAFLG        ;IF THE TEST FAILED ONLY WHEN
2601  005264  001403                      BEQ     KAD3            ;THE CONTROL REGISTER WAS SET
2602  005266  012737  177777  030756      MOV     #-1,CONFL2      ;SIGNAL A BAD CONTROL REGISTER.
2603  005274                  KAD3:                               ;DONE!!
2604
2605
2606                                      ;;*****************************************************************
2607                                      ;*TEST 7        CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 0 TEST
2608                                      ;*
2609                                      ;*THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS
2610                                      ;*OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS
2611                                      ;*MADE A HIT IN GROUP ONE; THEN ANOTHER ADDRESS, WHOSE
2612                                      ;*HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS
2613                                      ;*IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING
2614                                      ;*SELECTION OF GROUP ZERO; THEN SEE IF THE FIRST ADDRESS
2615                                      ;*IS STILL A HIT IN GROUP ONE; FINALLY TURN ON THE FORCE
2616                                      ;*MISS GROUP ZERO BIT AND SEE IF THE SECOND ADDRESS'
2617                                      ;*HIT IN GROUP ZERO CAN BE FORCED TO A MISS.
2618                                      ;*
2619                                      ;;*****************************************************************
2620  005274  000004          TST7:       SCOPE
2621  005276  012737  000040  001274      MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2622           000007          KD=$TN-1
2623                                                              ;SET THE SKAD REGISTER
2624  005304  012737  005624  030524      MOV     #TST10,SKAD     ;IN CASE THE TEST ABORTS.
2625
2626  005312  113737  001102  001224      MOVB    $TSTNM,$TMP0
2627  005320  012737  030400  000114      MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS.
2628
```

```
2629  005326  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2630  005330  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2631
2632  005332  012700  005622          K1D:    MOV     #KTMP2D,R0      ;DETERMINE THE TEST LOCATIONS.
2633  005336  042700  176003                  BIC     #176003,R0
2634  005342  010001                          MOV     R0,R1
2635  005344  062701  140000                  ADD     #TESTR1,R1
2636  005350  010137  001244                  MOV     R1,$TMP10
2637  005354  005037  001246                  CLR     $TMP11
2638  005360  010002                          MOV     R0,R2
2639  005362  062702  142000                  ADD     #TESTR2,R2
2640  005366  010237  001250                  MOV     R2,$TMP12
2641  005372  005037  001252                  CLR     $TMP13
2642
2643  005376  012737  000044  177746  K2D:    MOV     #S1M0,@#CONTRL  ;MAKE (R1) A HIT IN
2644  005404  005711                          TST     (R1)            ;GROUP GRM.
2645  005406  005711                          TST     (R1)
2646  005410  032737  000010  177752          BIT     #10,@#HITMIS
2647  005416  001007                          BNE     K3D
2648
2649                                                                  ;REPORT ERROR, UNABLE
2650  005420  012737  000001  001230          MOV     #1,$TMP2        ;GET A HIT IN GROUP GRM.
2651  005426  012737  000044  001232          MOV     #S1M0,$TMP3
2652  005434  104075                  1$:     ERROR   75
2653
2654  005436  012703  000030          K3D:    MOV     #S0M1,R3
2655  005442  042703  000017                  BIC     #17,R3
2656  005446  010337  177746                  MOV     R3,@#CONTRL     ;FORCE SELECT GROUP GRS.
2657  005452  005712                          TST     (R2)            ;MAKE (R2) A HIT IN GROUP
2658  005454  005712                          TST     (R2)            ;GRS.
2659  005456  032737  000010  177752          BIT     #10,@#HITMIS
2660  005464  001006                          BNE     K4D
2661                                                                  ;IF NOT, ERROR UNABLE TO
2662                                                                  ;GET A HIT IN GROUP 0
2663  005466  010337  001232                  MOV     R3,$TMP3
2664  005472  104076                  1$:     ERROR   76
2665  005474  012737  177777  030756          MOV     #-1,CONFL2
2666
2667  005502  005037  177746          K4D:    CLR     @#CONTRL        ;NOW MAKE SURE (R1) IS
2668  005506  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
2669  005510  005711                          TST     (R1)            ;STILL A HIT IN GROUP
2670  005512  032737  000010  177752          BIT     #10,@#HITMIS    ;1, THAT IS MAKE SURE
2671  005520  001010                          BNE     K5D             ;GROUP 1 WASN'T WRITTEN
2672                                                                  ;WHILE FORCE SELECTING
2673                                                                  ;GROUP GRS.
2674  005522  012737  000001  001230          MOV     #1,$TMP2
2675  005530  012737  000000  001232          MOV     #0,$TMP3
2676  005536  104077                  1$:     ERROR   77
2677  005540  000424                          BR      K6D
2678  005542  012703  000044          K5D:    MOV     #S1M0,R3        ;NOW SEE IF YOU CAN
2679  005546  042703  000063                  BIC     #63,R3          ;GET A MISS AT (R2)
2680  005552  010337  177746                  MOV     R3,@#CONTRL     ;BY FORCING MISSES
2681  005556  005712                          TST     (R2)            ;TO GRS.
2682  005560  032737  000010  177752          BIT     #10,@#HITMIS
2683  005566  001411                          BEQ     K6D             ;SHOULD BE A MISS.
2684                                                                  ;OTHERWISE ERROR!
```

```
                                    MOV     #0,$TMP2
                                    MOV     R3,$TMP3
                            1$:     ERROR   117
                                    MOV     #-1,CONFL2

                            K6D:    CLR     @#CONTRL
                                    BR      K7D

                            KTMP1D:.WORD    0
                            KTMP2D:.WORD    0

                            K7D:                                    ;DONE!


                            ;;*****************************************************************
                            ;*TEST 10       CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 1 TEST
                            ;*
                            ;*THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS
                            ;*OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS
                            ;*MADE A HIT IN GROUP ZERO; THEN ANOTHER ADDRESS, WHOSE
                            ;*HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS
                            ;*IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING
                            ;*SELECTION OF GROUP ONE; THEN SEE IF THE FIRST ADDRESS
                            ;*IS STILL A HIT IN GROUP ZERO; FINALLY TURN ON THE FORCE
                            ;*MISS GROUP ONE BIT AND SEE IF THE SECOND ADDRESS'
                            ;*HIT IN GROUP ONE CAN BE FORCED TO A MISS.
                            ;*
                            ;;*****************************************************************
                            TST10:  SCOPE
                                    MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                            KE=$TN-1
                                                            ;SET THE SKAD REGISTER
                                    MOV     #TST11,SKAD     ;IN CASE THE TEST ABORTS.

                                    MOVB    $TSTNM,$TMP0
                                    MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS.

                                    SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
                                    SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.

                            K1E:    MOV     #KTMP2E,R0      ;DETERMINE THE TEST LOCATIONS.
                                    BIC     #176003,R0
                                    MOV     R0,R1
                                    ADD     #TESTR1,R1
                                    MOV     R1,$TMP10
                                    CLR     $TMP11
                                    MOV     R0,R2
                                    ADD     #TESTR2,R2
                                    MOV     R2,$TMP12
                                    CLR     $TMP13

                            K2E:    MOV     #SOM1,@#CONTRL  ;MAKE (R1) A HIT IN
                                    TST     (R1)            ;GROUP GRM.
                                    TST     (R1)
                                    BIT     #10,@#HITMIS
                                    BNE     K3E
```

```
2741
2742
2743  005750  012737  000000  001230          MOV     #0,$TMP2        ;REPORT ERROR, UNABLE
2744  005756  012737  000030  001232          MOV     #SOM1,$TMP3     ;GET A HIT IN GROUP GRM.
2745  005764  104075                  1$:     ERROR   75
2746
2747  005766  012703  000044          K3E:    MOV     #S1M0,R3
2748  005772  042703  000017                  BIC     #17,R3
2749  005776  010337  177746                  MOV     R3,@#CONTRL     ;FORCE SELECT GROUP GRS.
2750  006002  005712                          TST     (R2)            ;MAKE (R2) A HIT IN GROUP
2751  006004  005712                          TST     (R2)            ;GRS.
2752  006006  032737  000010  177752          BIT     #10,@#HITMIS
2753  006014  001006                          BNE     K4E
2754                                                                  ;IF NOT, ERROR UNABLE TO
2755                                                                  ;GET A HIT IN GROUP 1
2756  006016  010337  001232                  MOV     R3,$TMP3
2757  006022  104076                  1$:     ERROR   76
2758  006024  012737  177777  030756          MOV     #-1,CONFL2
2759
2760  006032  005037  177746          K4E:    CLR     @#CONTRL        ;NOW MAKE SURE (R1) IS
2761  006036  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
2762  006040  005711                          TST     (R1)            ;STILL A HIT IN GROUP
2763  006042  032737  000010  177752          BIT     #10,@#HITMIS    ;0, THAT IS MAKE SURE
2764  006050  001010                          BNE     K5E             ;GROUP 0 WASN'T WRITTEN
2765                                                                  ;WHILE FORCE SELECTING
2766                                                                  ;GROUP GRS.
2767  006052  012737  000000  001230          MOV     #0,$TMP2
2768  006060  012737  000001  001232          MOV     #1,$TMP3
2769  006066  104077                  1$:     ERROR   77
2770  006070  000424                          BR      K6E
2771  006072  012703  000030          K5E:    MOV     #SOM1,R3        ;NOW SEE IF YOU CAN
2772  006076  042703  000063                  BIC     #63,R3          ;GET A MISS AT (R2)
2773  006102  010337  177746                  MOV     R3,@#CONTRL     ;BY FORCING MISSES
2774  006106  005712                          TST     (R2)            ;TO GRS.
2775  006110  032737  000010  177752          BIT     #10,@#HITMIS
2776  006116  001411                          BEQ     K6E             ;SHOULD BE A MISS.
2777                                                                  ;OTHERWISE ERROR!
2778  006120  012737  000001  001230          MOV     #1,$TMP2
2779  006126  010337  001232                  MOV     R3,$TMP3
2780  006132  104117                  1$:     ERROR   117
2781  006134  012737  177777  030756          MOV     #-1,CONFL2
2782
2783  006142  005037  177746          K6E:    CLR     @#CONTRL
2784  006146  000402                          BR      K7E
2785
2786  006150  000000                  KTMP1E:.WORD    0
2787  006152  000000                  KTMP2E:.WORD    0
2788
2789  006154                          K7E:                            ;DONE!
2790
2791
2792                                  ;;*****************************************************************
2793                                  ;*TEST 11        CACHE HIT/MISS REGISTER PATTERNS TEST
2794                                  ;*
2795                                  ;*THIS IS A TEST OF THE HIT/MISS REGISTER WHICH
2796                                  ;*FLOATS DIFFERENT PATTERNS OF HITS AND MISSES
```

```
                                        ;*THROUGH THAT REGISTER. THIS IS DONE FIRST WITH
                                        ;*BOTH GROUPS ENABLE; THEN WITH GROUP ZERO DISABLED
                                        ;*THAT IS FORCING SELECTION OF GROUP ONE AND FORCING
                                        ;*MISSES TO GROUP ZERO; FINALLY WITH GROUP ONE
                                        ;*DISABLED.
                                        ;*
                                        ;;*****************************************************************
006154  000004                    TST11:  SCOPE
006156  012737  000020  001274            MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000011              KC=$TN-1
                                                                    ;SET THE SKAD REGISTER
006164  012737  006764  030524            MOV     #TST12,SKAD     ;IN CASE THE TEST ABORTS.

006172  113737  001102  001224            MOVB    $TSTNM,$TMPO
006200  012737  030400  000114            MOV     #SPUR,@#CACHVEC

006206  104432                            SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
006210  104436                            SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
006212  005037  006646                    CLR     KCCON           ;TEST THE BOTH GROUPS
006216  012737  000002  006650            MOV     #2,KCFLG1       ;ENABLED CONDITION FIRST.
006224  012737  006240  001110  KC0:      MOV     #KC1,$LPERR
006232  012737  006654  006652            MOV     #KCTBL,KCPTR    ;KCPTR IS A POINTER TO
                                                                    ;THE TABLE OF 12-BIT PATTERNS
                                                                    ;WHICH WILL BE FLOATED
                                                                    ;THROUGH THE REGISTER.

006240  012701  140000          KC1:      MOV     #TESTR1,R1      ;MAKE THIS CODE MISSES
006244  012702  142000                    MOV     #TESTR2,R2      ;TO BOTH GROUPS!
006250  012700  001000                    MOV     #1000,R0
006254  012737  000030  177746  1$:       MOV     #SOM1,@#CONTRL
006262  005721                            TST     (R1)+
006264  012737  000044  177746            MOV     #S1M0,@#CONTRL
006272  005722                            TST     (R2)+
006274  077011                            SOB     R0,1$

006276  017702  000350                    MOV     @KCPTR,R2       ;GET THE HIT/MISS PATTERN
006302  012700  006362                    MOV     #KC3,R0         ;AND MAKE THE INSTRUCTIONS
006306  012701  000007                    MOV     #7,R1           ;BETWEEN KC3 AND KC9
006312  013737  006646  177746            MOV     KCCON,@#CONTRL  ;HITS AND MISSES SO THAT
006320  000403                            BR      KC2.5           ;WHEN THAT CODE IS EXECUTED
006322  006302                  KC2:      ASL     R2              ;THIS PATTERN WILL BE FLOATED
006324  103001                            BCC     KC2.5           ;THROUGH THE HIT/MISS REGISTER.
006326  005710                            TST     (R0)            ;MAKE (R0) A HIT!
006330  062700  000002          KC2.5:    ADD     #2,R0
006334  006302                            ASL     R2
006336  103001                            BCC     1$
006340  005710                            TST     (R0)            ;MAKE (R0) A HIT!
006342  062700  000006          1$:       ADD     #6,R0
006346  077113                            SOB     R1,KC2

006350  012705  177752                    MOV     #HITMIS,R5      ;NOW THAT THE HITS
006354  000402                            BR      KC3             ;AND MISSES HAVE BEEN
                                                                    ;APPROPRIATELY ESTABLISHED
                                                                    ;EXECUTE THE CODE AND
                                                                    ;CAUSE THE PATTERN TO FLOAT
                                                                    ;THROUGH THE HIT/MISS
```

```
                                                                    ;REGISTER.


                006356                          LOC=.               ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                006354                          LOC=-4&LOC
                006360                          LOC=LOC+4
                006360                          .=LOC

        006360  000000                          HALT
        006362  000240                  KC3:    NOP                         ;THE HALT'S HERE ARE NOT
        006364  000402                          BR      KC4                 ;EXECUTED, THEY ARE FILLERS.
        006366  000000                          HALT                        ;THE ADDRESS OF THE HIT AND
        006370  000000                          HALT                        ;MISS REGISTER IS IN R5.
        006372  011500                  KC4:    MOV     (R5),R0             ;NOTE THAT THE HIT/MISS
        006374  000402                          BR      KC5                 ;REGISTER IS READ EVERY
        006376  000000                          HALT                        ;TWO CYCLES AND SAVED IN
        006400  000000                          HALT                        ;A PROCESSOR GENERAL
        006402  011501                  KC5:    MOV     (R5),R1             ;PURPOSE REGISTER.
        006404  000402                          BR      KC6
        006406  000000                          HALT
        006410  000000                          HALT
        006412  011502                  KC6:    MOV     (R5),R2
        006414  000402                          BR      KC7
        006416  000000                          HALT
        006420  000000                          HALT
        006422  011503                  KC7:    MOV     (R5),R3
        006424  000402                          BR      KC8
        006426  000000                          HALT
        006430  000000                          HALT
        006432  011504                  KC8:    MOV     (R5),R4
        006434  000402                          BR      KC9
        006436  000000                          HALT
        006440  000000                          HALT
        006442  011505                  KC9:    MOV     (R5),R5             ;CAN SAVE PATTERN IN R5
                                                                            ;SINCE THE ADDRESS IS
                                                                            ;NO LONGER NEEDED.
        006444  042700  177774          KC10:   BIC     #177774,R0          ;GET THE PATTERNS READ
        006450  010037  006700                  MOV     R0,KCR0             ;FROM THE HIT/MISS REGISTER
        006454  042701  017760                  BIC     #17760,R1           ;INTO LOCATIONS KCR0
        006460  010137  006702                  MOV     R1,KCR1             ;THROUGH KCR5 SO THE
        006464  010237  006704                  MOV     R2,KCR2             ;GENERAL PURPOSE REGISTERS
        006470  010337  006706                  MOV     R3,KCR3             ;CAN BE USED FOR OTHER
        006474  010437  006710                  MOV     R4,KCR4             ;THINGS
        006500  010537  006712                  MOV     R5,KCR5

        006504  017701  000142          KC11:   MOV     @KCPTR,R1
        006510  005000                          CLR     R0
        006512  012702  000006                  MOV     #6,R2               ;PUT THE EXPECTED VALUES
        006516  012703  006714                  MOV     #KCE0,R3            ;IN KCE0 THROUGH KCE5!
        006522  073027  000002          KC12:   ASHC    #2,R0
        006526  042700  177700                  BIC     #177700,R0
        006532  010023                          MOV     R0,(R3)+
        006534  077206                          SOB     R2,KC12

        006536  012700  006700                  MOV     #KCR0,R0
        006542  012701  006714                  MOV     #KCE0,R1            ;MAKE SURE THE PATTERNS
```

```
                                                MOV     #6,R2           ;WHICH WERE READ FROM
                                        KC13:   CMP     (R0)+,(R1)+     ;THE HIT AND MISS REGISTER
                                                BEQ     KC14            ;MATCH THE EXPECTED
                                                JMP     KCERR           ;PATTERNS.
                                        KC14:   SOB     R2,KC13

                                        KC15:   ADD     #2,KCPTR                ;MOVE POINTER TO NEXT
                                                CMP     KCPTR,#KCTBLB   ;PATTERN AND IF ALL THE
                                                BEQ     1$              ;PATTERNS HAVEN'T BEEN
                                                JMP     KC1             ;TESTED GO TO KC1 TO TEST
                                                                        ;THIS NEXT PATTERN.
                                        1$:     DEC     KCFLG1          ;IF ALL THE PATERNS HAVE BEEN
                                                BPL     KC16            ;TESTED WITH THAT GROUP CONFIGURATION
                                                JMP     KCDONE          ;SO GO TO THE NEXT CONFIGURATION.
                                                                        ;OR DONE!!
                                        KC16:   BEQ     KC17
                                                MOV     #S1M0,KCCON     ;BOTH GROUPS ENABLED CONFIGURATION
                                                JMP     KC0             ;HAS BEEN TESTED SO NOW TEST GROUP
                                                                        ;ZERO DISABLED CONFIGURATION.
                                        KC17:   MOV     #S0M1,KCCON     ;BOTH GROUPS ENABLED AND GROUP ZERO
                                                                        ;DISABLED CONFIGURATIONS HAVE BOTH
                                                                        ;BEEN TESTED SO FINALLY TEST THE
                                                JMP     KC0             ;GROUP ONE DISABLED CONFIGURATION.


                                        KCCON:  .WORD   0               ;PATTERN BEING USED IN THE CONTROL REGISTER

                                        KCFLG1: .WORD   0               ;FLAG USED TO DETERMINE THE CONFIGURATION
                                                                        ;BEING TESTED.
                                        KCPTR:  .WORD   0               ;POINTER USED TO POINT TO THE PATTERN
                                                                        ;BEING TESTED IN KCTBL.


                                        KCTBL:  .WORD   0               ;PATTERNS WHICH ARE
                                                .WORD   002000          ;FLOATED THROUGH THE HIT/MISS
                                                .WORD   177760          ;REGISTER. ONLY THE UPPER
                                                .WORD   175760          ;12 BITS HAVE ANY SIGNIFICANCE!!
                                                .WORD   125240
                                                .WORD   146300
                                                .WORD   161600
                                                .WORD   100020
                                                .WORD   077740
                                        KCTBLB: .WORD   0


                                        KCR0:   .WORD   0               ;STORAGE FOR THE PATTERNS READ
                                        KCR1:   .WORD   0               ;OUT OF THE HUIT/MISS REGISTER.
                                        KCR2:   .WORD   0
                                        KCR3:   .WORD   0
                                        KCR4:   .WORD   0
                                        KCR5:   .WORD   0


                                        KCE0:   .WORD   0               ;EXPECTED VALUES FOR THE PATTERNS
                                        KCE1:   .WORD   0               ;READ FROM THE HIT/MISS REGISTER.
                                        KCE2:   .WORD   0
                                        KCE3:   .WORD   0
                                        KCE4:   .WORD   0
                                        KCE5:   .WORD   0
```

```
006730                          KCERR:                          ;REPORT THE PATTERN READ FROM THE
005730  013737  006646  001230          MOV     KCCON,$TMP2     ;HIT/MISS REGISTER WAS NOT THE EXPECTED
006736  104120                  1$:     ERROR   120             ;VALUE.
006740  012737  177777  030756          MOV     #-1,CONFL2
006746  012737  177777  030762          MOV     #-1,HIMFL2
006754  000137  006564                  JMP     KC15

006760  005037  177746          KCDONE: CLR     @#CONTRL        ;DONE!!


                                ;;*****************************************************************
                                ;*TEST 12       CACHE CONTROL AND HIT/MISS REGISTERS EVALUATION ROUTINE
                                ;*
                                ;*THIS IS NOT A TEST. THIS ROUTINE IS USED TO LOOK AT THE RESULTS
                                ;*OF TST5 THROUGH TST10, WHICH TESTED THE HIT/MISS REGISTER
                                ;*AND THE CONTROL REGISTER. THOSE TESTS HAVE SIGNALLED A BAD
                                ;*REGISTER USING THE FLAGS, CONFL2 AND HIMFL2, REPRESENTING THE
                                ;*CONTROL AND HIT/MISS REGISTERS RESPECTIVELY. IF ONE OF THESE
                                ;*REGISTERS WAS FOUND TO BE BAD THE FLAG SHOULD BE A -1. WHILE A
                                ;*ZERO FLAG INDICATES THAT THOSE TESTS FOUND THAT REGISTER
                                ;*FUNCTIONAL. THIS ROUTINE LOOKS AT THE FLAGS, CONFL2 AND HIMFL2,
                                ;*WHICH ARE CONSIDERED TO BE LOCAL AND TRANSFERS THE INDICATORS
                                ;*THEY CONTAIN TO THE GLOBAL FLAGS, CONFLG AND HIMFLG. THESE GLOBAL
                                ;*FLAGS ARE USED TO DESIGNATE TO THE REST OF THE PROGRAM THE FUNCTIONALITY
                                ;*OR DISFUNCTIONALITY OF THOSE REGISTERS.
                                ;*
                                ;;*****************************************************************
006764  000004                  TST12:  SCOPE
        000012                  KY=$TN-1
006766  005737  030756                  TST     CONFL2
006772  001403                          BEQ     KY1
006774  012737  177777  030742          MOV     #-1,CONFLG
007002  005737  030762          KY1:    TST     HIMFL2
007006  001403                          BEQ     KY2
007010  012737  177777  030746          MOV     #-1,HIMFLG
007016                          KY2:                            ;DONE


                                ;;*****************************************************************
                                ;*TEST 13       CACHE CONTROL LOGIC, 'RANDOM' FLIP FLOP TEST
                                ;*
                                ;*THIS IS A TEST OF THE 'RANDOM' CONTROL SIGNAL.
                                ;*A TEST IS MADE TO INSURE THAT THE 'RANDOM' FLIP-FLOP IS NOT STUCK
                                ;*AND IS TOGGLED ONCE FOR EVERY 'BUST' CYCLE INITIATED BY
                                ;*THE PROCESSOR. 'BUST' IS BUS START, A SIGNAL PRODUCED BY
                                ;*THE PROCESSOR WHENEVER IT THINKS IT IS ABOUT TO DO A MEMORY CYCLE.
                                ;*THE RANDOM FLIP FLOP IS USED IN THE CACHE TO DETERMINE WHICH
                                ;*GROUP TO WRITE IN THE EVENT OF A READ MISS CYCLE. IF THIS FLIP FLOP IS
                                ;*SET THEN GROUP ZERO IS WRITTEN; IF CLEAR THEN GROUP ONE IS WRITTEN.
                                ;*
                                ;;*****************************************************************
007016  000004                  TST13:  SCOPE
007020  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000013                  KF=$TN-1
                                                                ;SET THE SKAD REGISTER
007026  012737  007252  030524          MOV     #TST14,SKAD     ;IN CASE THE TEST ABORTS.
```

```
3021  007034  113737  001102  001224          MOVB    $TSTNM,$TMP0
3022  007042  012737  030400  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
3023
3024  007050  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3025  007052  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3026  007054  012700  007250          KF1:    MOV     #KFTMP2,R0      ;ESTABLISH A LOCATION FOR THE
3027                                                                  ;HITS TO BE MADE WHICH WON'T
3028                                                                  ;INTERFER WITH THE HITS CAUSED
3029                                                                  ;BY EXECUTION OF THIS CODE!
3030  007060  042700  176003                  BIC     #176003,R0
3031  007064  010001                          MOV     R0,R1
3032  007066  062701  140000                  ADD     #TESTR1,R1
3033  007072  010002                          MOV     R0,R2
3034  007074  062702  142000                  ADD     #TESTR2,R2
3035
3036  007100  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;MAKE THOSE TWO TEST LOCATIONS
3037  007106  005710                          TST     (R0)            ;(R1) AND (R2) MISSES IN BOTH
3038                                                                  ;GROUPS BY MAKING (R0) A HIT
3039                                                                  ;IN BOTH GROUPS.
3040
3041  007110  005710                          TST     (R0)
3042
3043                                                                  ;SEE IF REFERENCE ADDRESS
3044  007112  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
3045  007120  001006                          BNE     KF2
3046                                                                  ;IF NOT ERROR!
3047  007122  010037  001230                  MOV     R0,$TMP2
3048  007126  012737  000001  001226          MOV     #1,$TMP1
3049  007134  104001                          ERROR   1
3050
3051
3052
3053
3054  007136  012737  000030  177746  KF2:    MOV     #S0M1,@#CONTRL
3055  007144  005710                          TST     (R0)
3056
3057  007146  005710                          TST     (R0)
3058
3059                                                                  ;SEE IF REFERENCE ADDRESS
3060  007150  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
3061  007156  001006                          BNE     KF3
3062                                                                  ;IF NOT ERROR!
3063  007160  010037  001230                  MOV     R0,$TMP2
3064  007164  012737  000000  001226          MOV     #0,$TMP1
3065  007172  104001                          ERROR   1
3066
3067
3068
3069
3070  007174  005037  177746          KF3:    CLR     @#CONTRL        ;NOW THAT THE ADDRESSES (R1)
3071                                                                  ;AND (R2) ARE MISSES, REFERENCING
3072                                                                  ;THEM BOTH EACH IN CONSECUTIVE
3073                                                                  ;REFERNCES SHOULD CAUSE THEM BOTH
3074                                                                  ;TO BE MADE HITS IF THE RANDOM
3075                                                                  ;FLIP FLOP TOGGLES INBETWEEN THE
3076                                                                  ;TWO CYCLES!
```

```
                                                        ;NOTE THAT THESE TWO ADDRESSES
                                                        ;(R1) AND (R2) ARE SUCH THAT
                                                        ;IF THE RANDOM FLIP FLOP DIDN'T TOGGLE
                                                        ;THE HITS AT THE ADDRESSES
                                                        ;WOULD BE MUTUALLY EXCLUSIVE,
                                                        ;THAT IS BOTH THESE ADDRESSES
                                                        ;CAN'T BE HITS IN THE SAME GROUP!

007200  000240                  NOP                    ;FOR SCOPING WITH AN OSCILLOSCOPE!
007202  021112                  CMP     (R1),(R2)      ;HERE BOTH THE OPERAND FETCHES
                                                        ;SHOULD BE MISSES.
007204  021112                  CMP     (R1),(R2)      ;HERE BOTH THE OPERAND FETCHES
                                                        ;SHOULD BE HITS!
007206  013705  177752          MOV     @#HITMIS,R5
007212  005105                  COM     R5
007214  032705  000014          BIT     #14,R5         ;BOTH HITS ELSE ERROR.
007220  001411                  BEQ     KF4

007222  010137  001230          MOV     R1,$TMP2       ;REPORT THE ERROR.
007226  005037  001232          CLR     $TMP3
007232  010237  001234          MOV     R2,$TMP4
007236  005037  001236          CLR     $TMP5

007242  104121          1$:     ERROR   121
007244  000402          KF4:    BR      KF5

007246  000000          KFTMP1: .WORD   0              ;USED TO DETERMINE THE TEST
007250  000000          KFTMP2: .WORD   0              ;ADDRESSES.

007252                  KF5:                           ;DONE!

                        ;;*****************************************************************
                        ;*TEST 14       CACHE MAINTENANCE REGISTER COUNT PATTERN TEST
                        ;*
                        ;*THIS TEST RUNS A COUNT PATTERN THROUGH THE MAINTENANCE REGISTER'S
                        ;*BITS 15 TO 4. THIS IS DONE TO INSURE THAT THESE BITS ARE SETABLE
                        ;*AND THAT THE DATA PATH TO THE REGISTERS IS VIABLE. MISSES ARE FORCED
                        ;*TO BOTH GROUPS SO THAT NO CACHE DATA OR ADDRESS MEMORY
                        ;*ERRORS SHOULD OCCUR. ALSO ANY CYCLES DONE TO MAIN MEMORY
                        ;*ARE INSURED, BY PROPER SELECTION OF INSTRUCTIONS, TO RETURN
                        ;*DATA WITH THE PARITY BITS ON SO AS TO NOT CAUSE MAIN MEMORY PARITY
                        ;*ERRORS BY SETTING THE MAIN MEMORY MAINTENANCE FUNCTION WHICH WOULD
                        ;*EFFECTIVELY FORCE THE PARITY BITS READ FROM MAIN MEMORY TO A
                        ;*ONE. SINCE THESE PARITY ARE ALREADY ONES, NO ERRORS SHOULD OCCUR.
                        ;*
                        ;;*****************************************************************
007252  000004          TST14:  SCOPE
007254  012737  000020  001274          MOV     #20,$TIMES     ;;DO 20 ITERATIONS
        000014          MA=$TN-1
                                                        ;SET THE SKAD REGISTER
007262  012737  007534  030524          MOV     #TST15,SKAD    ;IN CASE THE TEST ABORTS.

007270  113737  001102  001224          MOVB    $TSTNM,$TMP0

007276  104432                  SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
007300  104434                  SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
```

```
3133  007302  012737  007436  000114          MOV     #MAERR,@#CACHVEC        ;IN CASE AN ERROR OCCURS WHILE
3134                                                                          ;RUNNING A COUNT PATTERN
3135                                                                          ;THROUGH THE MAINTENANCE
3136                                                                          ;REGISTER SET UP THE PARITY ERROR
3137                                                                          ;TRAP VECTOR; NOTE THAT NO ERRORS
3138                                                                          ;SHOULD OCCUR IF THIS REGISTER
3139                                                                          ;AND THE PARITY LOGIC IS FUNCTIONING
3140                                                                          ;PROPERLY!
3141  007310  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
3142
3143  007316  012701  177750                  MOV     #MAINT,R1
3144  007322  005004                          CLR     R4
3145  007324  012737  007336  001110          MOV     #MA1,$LPERR
3146  007332  012700  170000                  MOV     #170000,R0
3147
3148  007336  000240                  MA1:    NOP                             ;NOTE, THE CODE IN THIS ARE
3149  007340  010411                          MOV     R4,(R1)                 ;MA1 THROUGH MA2, ASSEMBLES TO
3150  007342  011102                          MOV     (R1),R2                 ;MACHINE CODE WHICH WILL
3151  007344  005011                          CLR     (R1)                    ;HAVE THE PARITY BITS ON, 1'S!
3152                                                                          ;THE PATTERN IS LOADDED INTO THE
3153                                                                          ;MAINETENANCE REGISTER, READ BACK
3154                                                                          ;AND THE MAINTENANCE REGISTER
3155                                                                          ;IS CLEARED.
3156  007346  030011                          BIT     R0,(R1)                 ;SEE IF ANY OF THE HIGH ORDER
3157                                                                          ;FOUR BITS, 15 TO 12,
3158                                                                          ;THE BITS WHICH CONTROL THE
3159                                                                          ;MAIN MEMORY DATA PARITY MAINTENANCE
3160                                                                          ;FUNCTION ARE STUCK ON.
3161  007350  001402                          BEQ     .+6                     ;IF SO, THEN ALL THAT CAN
3162  007352  000000                          HALT                            ;BE DONE IS TO HALT!!!!!
3163                                                                          ;FOR IF CONTROL IS PASSED TO
3164                                                                          ;ANY OTHER PART OF THIS PROGRAM
3165                                                                          ;THERE WOULD BE NO CONTROL
3166                                                                          ;OVER WHAT KIND OF DATA WOULD
3167                                                                          ;BE READ FROM MAIN MEMORY AND
3168                                                                          ;MAIN MEMORY DATA PARITY ERRORS
3169                                                                          ;WOULD BE LIKELY TO OCCUR.
3170  007354  000240                  MA2:    NOP
3171
3172  007356  011105                          MOV     (R1),R5                 ;SEE IF ANY OF THE LOW ORDER
3173  007360  001410                          BEQ     MA3                     ;BITS, 11 THROUGH 0, ARE STUCK
3174                                                                          ;AT ONE.
3175                                                                          ;IF SO REPORT THE ERROR.
3176  007362  010437  001230                  MOV     R4,$TMP2
3177  007366  010537  001232                  MOV     R5,$TMP3
3178  007372  104122                  1$:     ERROR   122
3179  007374  012737  177777  030744          MOV     #-1,MANFLG              ;??????????????GO ON???????????
3180
3181  007402  020402                  MA3:    CMP     R4,R2                   ;SEE IF THE PATTERN WRITTEN MATCHES
3182  007404  001410                          BEQ     MA4                     ;THE PATTERN READ.
3183
3184                                                                          ;IF NOT REPORT THE ERROR.
3185  007406  010437  001230                  MOV     R4,$TMP2
3186  007412  010237  001232                  MOV     R2,$TMP3
3187  007416  104123                  1$:     ERROR   123
3188  007420  012737  177777  030760          MOV     #-1,MANFL2
```

```
3189
3190  007426  062704  000020          MA4:    ADD     #20,R4          ;INCREMENT THE COUNT PATTERN.
3191  007432  001341                          BNE     MA1
3192  007434  000432                          BR      MADONE
3193
3194  007436                          MAERR:                          ;TRAP TO HERE IN THE EVENT
3195                                                                  ;THAT A PARITY ERROR OCCURS
3196                                                                  ;WHILE RUNNING THIS COUNT
3197                                                                  ;PATTERN TEST.
3198  007436  032737  000400  177744        BIT     #400,@#MEMERR   ;SEE IF THE ERROR WAS A MAINTENANCE
3199  007444  001005                        BNE     MAERR1          ;ERROR, CAUSED BY A MAINTENANCE
3200                                                                  ;FUNCTION. IF NOT GO TO THE
3201  007446  012737  030400  000114        MOV     #SPUR,@#CACHVEC ;SPUR ROUTINE WHICH HANDLES SUCH UNEXPECTED
3202  007454  000137  030400                JMP     SPUR            ;ERRORS.
3203
3204  007460  013737  177744  001234  MAERR1: MOV     @#MEMERR,$TMP4  ;IF THE ERROR WAS CAUSED BY A
3205  007466  013737  177740  001226        MOV     @#LOADRS,$TMP1  ;MAINT FUNCTION THEN REPORT THE
3206  007474  013737  177742  001230        MOV     @#HIADRS,$TMP2  ;FAILURE OF THAT REGISTER.
3207  007502  012637  001232                MOV     (SP)+,$TMP3
3208  007506  005726                        TST     (SP)+
3209  007510  104124                  1$:   ERROR   124
3210  007512  012737  177777  030760        MOV     #-1,MANFL2
3211
3212  007520  000742                        BR      MA4             ;RETURN TO THE TEST.
3213
3214  007522  005037  177746  MADONE: CLR     @#CONTRL        ;DONE
3215  007526  012737  030400  000114        MOV     #SPUR,@#CACHVEC
3216
3217
3218
3219
3220                                  ;;**************************************************************
3221                                  ;*TEST 15       CACHE MAINTENANCE AND ERROR REGISTERS TEST 1
3222                                  ;*
3223                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY
3224                                  ;*ERROR ON THE MAIN MEMORY ADDRESS AND CONTROL LINES, AND ALSO A TEST
3225                                  ;*OF THE ERROR REGISTER'S ABILITY TO APPROPRIATELY SET TO 104402. THE
3226                                  ;*REFERENCE CAUSING THIS ERROR WILL BE MADE FROM THE CPU DIRECTLY TO
3227                                  ;*THE CACHE.
3228                                  ;*
3229                                  ;;**************************************************************
3230  007534  000004          TST15:  SCOPE
3231  007536  012737  000040  001274        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3232          000015          MAB=$TN-1
3233                                                                  ;SET THE SKAD REGISTER
3234  007544  012737  010032  030524        MOV     #TST16,SKAD     ;IN CASE THE TEST ABORTS.
3235
3236  007552  113737  001102  001224        MOVB    $TSTNM,$TMPO
3237
3238  007560  104430                        SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3239  007562  104432                        SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3240  007564  104434                        SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3241  007566  104436                        SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3242  007570  012737  007640  000114        MOV     #MABRRO,@#CACHVEC       ;SET UP FOR THE ERROR.
3243
3244  007576  012704  000002                MOV     #2,R4           ;THIS IS THE PATTERN THAT WILL
```

```
3245  007602  012702  177750            MOV     #MAINT,R2       ;BE PUT IN THE MAINTENANCE REG.
3246  007606  012737  000014  177746    MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
3247
3248  007614  000240                    NOP                     ;FOR SCOPING.
3249  007616  010412                    MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
3250  007620  005012                    CLR     (R2)            ;THE REFERENCE WHICH FETCHES
3251                                                            ;THIS INSTRUCTION SHOULD
3252                                                            ;CAUSE THE ABORT!
3253
3254  007622                    MAB2:                           ;NO ABORT OCCURRED REPORT THE ERROR
3255  007622  010437  001230            MOV     R4,$TMP2
3256  007626  104127            1$:     ERROR   127
3257  007630  012737  177777  030760    MOV     #-1,MANFL2
3258  007636  000474                    BR      MABDON
3259
3260  007640  022737  104402  177744  MABRR0: CMP     #104402,@#MEMERR        ;WHEN THE TRAP IS MADE TO THIS LOCATION
3261  007646  001036                    BNE     MABRR4          ;MAKE SURE THE ERROR REGISTER IS
3262                                                            ;SET CORRECTLY. IF NOT GO TO MABRR4.
3263  007650  022626            MABRR1: CMP     (SP)+,(SP)+     ;OTHERWISE RESET THE STACK.
3264  007652  012737  177777  177744  MABR15: MOV     #-1,@#MEMERR    ;ATTEMPT TO CLEAR THE ERROR REGISTER.
3265  007660  005737  177744            TST     @#MEMERR
3266  007664  001416                    BEQ     MABRR3
3267
3268  007666                    MABRR2:                         ;REPORT ERROR REGISTER WON'T CLEAR!
3269  007666  013737  177740  001230    MOV     @#LOADRS,$TMP2
3270  007674  013737  177742  001232    MOV     @#HIADRS,$TMP3
3271  007702  013737  177744  001234    MOV     @#MEMERR,$TMP4
3272  007710  104130            1$:     ERROR   130
3273  007712  012737  177777  030740    MOV     #-1,MMRFLG
3274  007720  000443                    BR      MABDON
3275
3276  007722  022737  177740  177740  MABRR3: CMP     #177740,@#LOADRS        ;MAKE SURE THE ADDRESS
3277  007730  001356                    BNE     MABRR2          ;REGISTER RESET.
3278  007732  022737  000003  177742    CMP     #3,@#HIADRS
3279  007740  001352                    BNE     MABRR2
3280  007742  000432                    BR      MABDON
3281
3282  007744                    MABRR4:                         ;REPORT ERROR REGISTER NOT SET CORRECTLY!!
3283  007744  012637  001230            MOV     (SP)+,$TMP2
3284  007750  005726                    TST     (SP)+
3285  007752  013737  177740  001232    MOV     @#LOADRS,$TMP3
3286  007760  013737  177742  001234    MOV     @#HIADRS,$TMP4
3287  007766  012737  000002  001236    MOV     #2,$TMP5
3288  007774  012737  104402  001240    MOV     #104402,$TMP6
3289  010002  013737  177744  001242    MOV     @#MEMERR,$TMP7
3290  010010  104131            1$:     ERROR   131
3291  010012  012737  177777  030760    MOV     #-1,MANFL2
3292  010020  012737  177777  030754    MOV     #-1,MMRFL2
3293  010026  000711                    BR      MABR15          ;GO SEE IF THE ERROR REGISTER
3294                                                            ;CAN BE CLEARED.
3295  010030  104416            MABDON: RSET                    ;DONE!!
3296
3297
3298                            ;;*****************************************************************
3299                            ;*TEST 16       CACHE MAINTENANCE AND ERROR REGISTERS TEST 2
3300                            ;*
```

```
                                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
                                        ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE,
                                        ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
                                        ;*
                                        ;;*****************************************************************
010032  000004                        TST16:  SCOPE
010034  012737  000040  001274                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000016                  MB=$TN-1
                                                                        ;SET THE SKAD REGISTER
010042  012737  010350  030524          MOV     #TST17,SKAD             ;IN CASE THE TEST ABORTS.

010050  113737  001102  001224          MOVB    $TSTNM,$TMPO

010056  104430                         SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
010060  104432                         SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
010062  104434                         SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
010064  104436                         SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
010066  012737  010146  000114          MOV     #MBERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
010074  012704  010000                  MOV     #10000,R4               ;PATERN TO BE PUT INTO THE
010100  012702  177750                  MOV     #MAINT,R2               ;MAINTENANCE REGISTER.
010104  012737  000014  177746          MOV     #MDM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
010112  000402                          BR      MB1

        010114                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        010114                          LOC=-4&LOC
        010120                          LOC=LOC+4
        010120                          .=LOC

010120  000240                  MB1:    NOP
010122  010412                          MOV     R4,(R2)                 ;SET THE MAINTENANCE REGISTER.
010124  005701                  MB2:    TST     R1                      ;THIS IS A DUMMY INSTRUCTION
                                                                        ;WITH THE APPROPRIATE PARITY
                                                                        ;WHOSE FETCH WILL CAUSE THE ERROR.
010126  005012                          CLR     (R2)

010130                          MB3:                                    ;REPORT ERROR.  MAINTENANCE
010130  010437  001230                  MOV     R4,$TMP2                ;FUNCTION FAILED TO
                                                                        ;CAUSE ERROR.
010134  104127                  1$:     ERROR   127
010136  012737  177777  030760  MOV     #-1,MANFL2
010144  000500                          BR      MBDONE

010146  022737  104404  177744  MBERRO: CMP     #104404,@#MEMERR                ;DID THE ERROR REGISTER
010154  001042                          BNE     69$                     ;SET PROPERLY?

010156  022626                  64$:    CMP     (SP)+,(SP)+             ;RESET THE STACK
010160  005037  177572          65$:    CLR     @#MMR0
010164  005037  172516                  CLR     @#MMR3
010170  012737  177777  177744          MOV     #-1,@#MEMERR            ;TRY TO CLEAR THE ERROR
010176  005737  177744                  TST     @#MEMERR                ;REGISTER.
010202  001416                          BEQ     68$

010204                          66$:                                    ;ERROR REGISTER WON'T
010204  013737  177740  001230          MOV     @#LOADRS,$TMP2          ;CLEAR
010212  013737  177742  001232          MOV     @#HIADRS,$TMP3
010220  013737  177744  001234          MOV     @#MEMERR,$TMP4
```

```
                                67$:    ERROR   130
010226  104130
010230  012737  177777  030740        MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
010236  000443                        BR      MBDONE

010240  022737  177740  177740  68$:  CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
010246  001356                        BNE     66$             ;UNLOCKED.
010250  022737  000003  177742        CMP     #3,@#HIADRS
010256  001352                        BNE     66$
010260  000432                        BR      MBDONE

010262                          69$:                            ;REPORT ERROR REGISTER
010262  012637  001230                MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
010266  005726                        TST     (SP)+           ;RESET THE STACK.
010270  013737  177740  001232        MOV     @#LOADRS,$TMP3
010276  013737  177742  001234        MOV     @#HIADRS,$TMP4
010304  012737  010000  001236        MOV     #10000,$TMP5
010312  012737  104404  001240        MOV     #104404,$TMP6
010320  013737  177744  001242        MOV     @#MEMERR,$TMP7

010326  104131                  70$:  ERROR   131
010330  012737  177777  030760        MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
010336  012737  177777  030754        MOV     #-1,MMRFL2
010344  000705                        BR      65$
010346  104416                MBDONE: RSET

                        ;;**************************************************************
                        ;*TEST 17        CACHE MAINTENANCE AND ERROR REGISTERS TEST 3
                        ;*
                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
                        ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S HIGH BYTE,
                        ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
                        ;*
                        ;;**************************************************************
010350  000004         TST17:  SCOPE
010352  012737  000040  001274        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000017          MC=$TN-1
                                                        ;SET THE SKAD REGISTER
010360  012737  010664  030524        MOV     #TST20,SKAD     ;IN CASE THE TEST ABORTS.

010366  113737  001102  001224        MOVB    $TSTNM,$TMP0

010374  104430                       SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
010376  104432                       SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
010400  104434                       SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
010402  104436                       SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
010404  012737  010462  000114        MOV     #MCERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
010412  012704  020000                MOV     #20000,R4       ;PATTERN TO BE USED IN THE
010416  012702  177750                MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
010422  012737  000014  177746        MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
010430  000401                        BR      MC1

        010432                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        010430                          LOC=-4&LOC
        010434                          LOC=LOC+4
        010434                          .=LOC
```

```
3413
3414  010434  000240                    MC1:    NOP
3415  010436  010412                            MOV     R4,(R2)          ;SET THE MAINTENANCE REGISTER.
3416  010440  005701                    MC2:    TST     R1               ;THE FETCH OF THIS INSTRUCTION
3417                                                                       ;SHOULD CAUSE THE ABORT.
3418  010442  005012                            CLR     (R2)
3419
3420  010444                            MC3:                             ;REPORT ERROR.  MAINTENANCE
3421  010444  010437  001230                    MOV     R4,$TMP2         ;FUNCTION FAILED TO
3422                                                                       ;CAUSE ERROR.
3423  010450  104127                    1$:     ERROR   127
3424  010452  012737  177777  030760    MOV     #-1,MANFL2
3425  010460  000500                            BR      MCDONE
3426
3427  010462  022737  104404  177744    MCERR0: CMP     #104404,@#MEMERR         ;DID THE ERROR REGISTER
3428  010470  001042                            BNE     69$              ;SET PROPERLY?
3429
3430  010472  022626                    64$:    CMP     (SP)+,(SP)+      ;RESET THE STACK
3431  010474  005037  177572            65$:    CLR     @#MMR0
3432  010500  005037  172516                    CLR     @#MMR3
3433  010504  012737  177777  177744            MOV     #-1,@#MEMERR     ;TRY TO CLEAR THE ERROR
3434  010512  005737  177744                    TST     @#MEMERR         ;REGISTER.
3435  010516  001416                            BEQ     68$
3436
3437  010520                            66$:                             ;ERROR REGISTER WON'T
3438  010520  013737  177740  001230            MOV     @#LOADRS,$TMP2   ;CLEAR
3439  010526  013737  177742  001232            MOV     @#HIADRS,$TMP3
3440  010534  013737  177744  001234            MOV     @#MEMERR,$TMP4
3441
3442  010542  104130                    67$:    ERROR   130
3443  010544  012737  177777  030740            MOV     #-1,MMRFLG       ;SIGNAL BAD REGISTER
3444  010552  000443                            BR      MCDONE
3445
3446  010554  022737  177740  177740    68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3447  010562  001356                            BNE     66$              ;UNLOCKED.
3448  010564  022737  000003  177742            CMP     #3,@#HIADRS
3449  010572  001352                            BNE     66$
3450  010574  000432                            BR      MCDONE
3451
3452  010576                            69$:                             ;REPORT ERROR REGISTER
3453  010576  012637  001230                    MOV     (SP)+,$TMP2      ;NOT SET AS EXPECTED.
3454  010602  005726                            TST     (SP)+            ;RESET THE STACK.
3455  010604  013737  177740  001232            MOV     @#LOADRS,$TMP3
3456  010612  013737  177742  001234            MOV     @#HIADRS,$TMP4
3457  010620  012737  020000  001236            MOV     #20000,$TMP5
3458  010626  012737  104404  001240            MOV     #104404,$TMP6
3459  010634  013737  177744  001242            MOV     @#MEMERR,$TMP7
3460
3461  010642  104131                    70$:    ERROR   131
3462  010644  012737  177777  030760            MOV     #-1,MANFL2       ;SIGNAL BAD REGISTER
3463  010652  012737  177777  030754            MOV     #-1,MMRFL2
3464  010660  000705                            BR      65$
3465  010662  104416                    MCDONE: RSET
3466
3467                                    ;;*****************************************************************
3468                                    ;*TEST 20        CACHE MAINTENANCE AND ERROR REGISTERS TEST 4
```

```
                                        ;*
                                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
                                        ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE,
                                        ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
                                        ;*
                                        ;;*****************************************************************
010664  000004                        TST20:  SCOPE
010666  012737  000040  001274                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000020                  MD=$TN-1
                                                                        ;SET THE SKAD REGISTER
010674  012737  011204  030524          MOV     #TST21,SKAD     ;IN CASE THE TEST ABORTS.

010702  113737  001102  001224          MOVB    $TSTNM,$TMP0

010710  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
010712  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
010714  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
010716  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
010720  012737  011002  000114          MOV     #MDERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
010726  012704  040000                  MOV     #40000,R4       ;PATTERN TO BE PUT IN THE
010732  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
010736  012737  000014  177746          MOV     #MDM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
010744  000402                          BR      MD1

                010746                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                010744                  LOC=-4&LOC
                010750                  LOC=LOC+4
                010750                  .=LOC

010750  000240                          NOP
010752  000240                  MD1:    NOP
010754  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
010756  005701                  MD2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
                                                                ;SHOULD CAUSE THE MAIN MEMORY
                                                                ;DATA PARITY ABORT.
010760  005012                          CLR     (R2)
010762  000240                          NOP

010764                          MD3:                            ;REPORT ERROR.  MAINTENANCE
010764  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
010770  104127                  1$:     ERROR   127
010772  012737  177777  030760  MOV     #-1,MANFL2
011000  000500                          BR      MDDONE

011002  022737  104410  177744  MDERRO: CMP     #104410,@#MEMERR        ;DID THE ERROR REGISTER
011010  001042                          BNE     69$             ;SET PROPERLY?

011012  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
011014  005037  177572          65$:    CLR     @#MMR0
011020  005037  172516                  CLR     @#MMR3
011024  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
011032  005737  177744                  TST     @#MEMERR        ;REGISTER.
011036  001416                          BEQ     68$

011040                          66$:                            ;ERROR REGISTER WON'T
```

```
                                        MOV     @#LOADRS,$TMP2  ;CLEAR
                                        MOV     @#HIADRS,$TMP3
                                        MOV     @#MEMERR,$TMP4

                        67$:    ERROR   130
                                MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
                                BR      MDDONE

                        68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
                                BNE     66$             ;UNLOCKED.
                                CMP     #3,@#HIADRS
                                BNE     66$
                                BR      MDDONE

                        69$:                            ;REPORT ERROR REGISTER
                                MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
                                TST     (SP)+           ;RESET THE STACK.
                                MOV     @#LOADRS,$TMP3
                                MOV     @#HIADRS,$TMP4
                                MOV     #40000,$TMP5
                                MOV     #104410,$TMP6
                                MOV     @#MEMERR,$TMP7

                        70$:    ERROR   131
                                MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
                                MOV     #-1,MMRFL2
                                BR      65$
                        MDDONE: RSET

                        ;;*****************************************************************
                        ;*TEST 21       CACHE MAINTENANCE AND ERROR REGISTERS TEST 5
                        ;*
                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
                        ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S HIGH BYTE,
                        ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
                        ;*
                        ;;*****************************************************************
                        TST21:  SCOPE
                                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                        ME=$TN-1
                                                        ;SET THE SKAD REGISTER
                                MOV     #TST22,SKAD     ;IN CASE THE TEST ABORTS.

                                MOVB    $TSTNM,$TMP0

                                SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
                                SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
                                SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
                                SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
                                MOV     #MEERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
                                MOV     #100000,R4      ;PATTERN TO BE PUT IN THE
                                MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
                                MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
                                BR      ME1

                                LOC=..                  ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
```

```
                011264                          LOC=-4&LOC
                011270                          LOC=LOC+4
                011270                          .=LOC

011270  000240                                 NOP
011272  000240                          ME1:    NOP
011274  010412                                  MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
011276  005701                          ME2:    TST     R1              ;THE FETCH OF THIS INSTRUSTION
                                                                        ;SHOULD CAUSE THE ABORT.
011300  005012                                  CLR     (R2)
011302  000240                                  NOP

011304                          ME3:                                    ;REPORT ERROR.  MAINTENANCE
011304  010437  001230                          MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                        ;CAUSE ERROR.
011310  104127                  1$:     ERROR   127
011312  012737  177777  030760  MOV     #-1,MANFL2
011320  000500                          BR      MEDONE

011322  022737  104410  177744  MEERRO: CMP     #104410,@#MEMERR        ;DID THE ERROR REGISTER
011330  001042                          BNE     69$             ;SET PROPERLY?

011332  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
011334  005037  177572          65$:    CLR     @#MMR0
011340  005037  172516                  CLR     @#MMR3
011344  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
011352  005737  177744                  TST     @#MEMERR        ;REGISTER.
011356  001416                          BEQ     68$

011360                          66$:                            ;ERROR REGISTER WON'T
011360  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
011366  013737  177742  001232          MOV     @#HIADRS,$TMP3
011374  013737  177744  001234          MOV     @#MEMERR,$TMP4

011402  104130                  67$:    ERROR   130
011404  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
011412  000443                          BR      MEDONE

011414  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
011422  001356                          BNE     66$             ;UNLOCKED.
011424  022737  000003  177742          CMP     #3,@#HIADRS
011432  001352                          BNE     66$
011434  000432                          BR      MEDONE

011436                          69$:                            ;REPORT ERROR REGISTER
011436  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
011442  005726                          TST     (SP)+           ;RESET THE STACK.
011444  013737  177740  001232          MOV     @#LOADRS,$TMP3
011452  013737  177742  001234          MOV     @#HIADRS,$TMP4
011460  012737  100000  001236          MOV     #100000,$TMP5
011466  012737  104410  001240          MOV     #104410,$TMP6
011474  013737  177744  001242          MOV     @#MEMERR,$TMP7

011502  104131                  70$:    ERROR   131
011504  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
011512  012737  177777  030754          MOV     #-1,MMRFL2
```

```
011520  000705                          BR      65$
011522  104416                  MEDONE: RSET

                                ;;*****************************************************************
                                ;*TEST 22        CACHE MAINTENANCE AND ERROR REGISTERS TEST 6
                                ;*
                                ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
                                ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE,
                                ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
                                ;*
                                ;;*****************************************************************
011524  000004                  TST22:  SCOPE
011526  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000022          MF=$TN-1
                                                                ;SET THE SKAD REGISTER
011534  012737  012040  030524          MOV     #TST23,SKAD     ;IN CASE THE TEST ABORTS.

011542  113737  001102  001224          MOVB    $TSTNM,$TMP0
011550  012737  011636  000114          MOV     #MFERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
011556  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE LOADED INTO THE
011562  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
011566  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
011574  012705  011616                  MOV     #MF2,R5         ;A REFERENCE TO THIS ADDRESS
                                                                ;WILL CAUSE A PARITY TRAP BECAUSE
                                                                ;THE OTHER WORD IN THE PAIR
                                                                ;WILL HAVE THE APPROPRIATE
                                                                ;PARITY TO CAUSE THE MAINTENANCE
                                                                ;FUNCTION WHICH WILL BE SET
                                                                ;TO FORCE THE ERROR.
011600  000401                          BR      MF1

                        011602          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                        011600          LOC=-4&LOC
                        011604          LOC=LOC+4
                        011604          .=LOC

011604  000240                  MF1:    NOP
011606  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
011610  021502                          CMP     (R5),R2         ;THIS REFERENCE TO (R5) WILL CAUSE A
011612  005012                          CLR     (R2)            ;PARITY TRAP SINCE THE OTHER IN THAT
                                                                ;PAIR WILL CAUSE A PARITY ERROR.
011614  005701                          TST     R1              ;THIS WORD WILL CAUSE THE ERROR
011616  000240                  MF2:    NOP                     ;WHEN THIS WORD IS REFERENCED.

011620                          MF3:                            ;REPORT ERROR.  MAINTENANCE
011620  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
011624  104127                  1$:     ERROR   127
011626  012737  177777  030760  MOV     #-1,MANFL2
011634  000500                  BR      MFDONE

011636  022737  004404  177744  MFERR0: CMP     #4404,@#MEMERR  ;DID THE ERROR REGISTER
011644  001042                  BNE     69$             ;SET PROPERLY?

011646  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
011650  005037  177572          65$:    CLR     @#MMR0
```

```
3693  011654  005037  172516              CLR     @#MMR3
3694  011660  012737  177777  177744      MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3695  011666  005737  177744              TST     @#MEMERR        ;REGISTER.
3696  011672  001416                      BEQ     68$
3697
3698  011674                      66$:                            ;ERROR REGISTER WON'T
3699  011674  013737  177740  001230      MOV     @#LOADRS,$TMP2  ;CLEAR
3700  011702  013737  177742  001232      MOV     @#HIADRS,$TMP3
3701  011710  013737  177744  001234      MOV     @#MEMERR,$TMP4
3702
3703  011716  104130              67$:    ERROR   130
3704  011720  012737  177777  030740      MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3705  011726  000443                      BR      MFDONE
3706
3707  011730  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3708  011736  001356                      BNE     66$             ;UNLOCKED.
3709  011740  022737  000003  177742          CMP     #3,@#HIADRS
3710  011746  001352                      BNE     66$
3711  011750  000432                      BR      MFDONE
3712
3713  011752                      69$:                            ;REPORT ERROR REGISTER
3714  011752  012637  001230              MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3715  011756  005726                      TST     (SP)+           ;RESET THE STACK.
3716  011760  013737  177740  001232      MOV     @#LOADRS,$TMP3
3717  011766  013737  177742  001234      MOV     @#HIADRS,$TMP4
3718  011774  012737  010000  001236      MOV     #10000,$TMP5
3719  012002  012737  004404  001240      MOV     #4404,$TMP6
3720  012010  013737  177744  001242      MOV     @#MEMERR,$TMP7
3721
3722  012016  104131              70$:    ERROR   131
3723  012020  012737  177777  030760      MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3724  012026  012737  177777  030754      MOV     #-1,MMRFL2
3725  012034  000705                      BR      65$
3726  012036  104416              MFDONE: RSET
3727
3728                                      ;;*****************************************************************
3729                                      ;*TEST 23        CACHE MAINTENANCE AND ERROR REGISTERS TEST 7
3730                                      ;*
3731                                      ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
3732                                      ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE,
3733                                      ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
3734                                      ;*
3735                                      ;;*****************************************************************
3736  012040  000004              TST23:  SCOPE
3737  012042  012737  000040  001274      MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3738          000023              MG=$TN-1
3739                                                              ;SET THE SKAD REGISTER
3740  012050  012737  012360  030524      MOV     #TST24,SKAD     ;IN CASE THE TEST ABORTS.
3741
3742  012056  113737  001102  001224      MOVB    $TSTNM,$TMP0
3743
3744  012064  104430                      SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3745  012066  104432                      SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3746  012070  104434                      SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3747  012072  104436                      SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3748  012074  012704  040000              MOV     #40000,R4       ;THIS PATTERN WILL BE PUT IN THE
```

```
 3749  012100  012704  177750                MOV     #MAINT,R4       ;MAINTENANCE REGISTER.
 3750  012104  012737  012156  000114        MOV     #MGERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
 3751  012112  012737  000014  177746        MOV     #MDM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
 3752  012120  000401                        BR      MG1
 3753
 3754          012122                        LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 3755          012120                        LOC=-4&LOC
 3756          012124                        LOC=LOC+4
 3757          012124                        .=LOC
 3758
 3759  012124  000240                MG1:    NOP
 3760  012126  010412                        MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
 3761  012130  000240                        NOP                     ;THE REFERENCE TO THIS NOP
 3762  012132  005701                MG2:    TST     R1              ;SHOULD CAUSE A PARITY ERROR TO OCCUR AT
 3763                                                                ;MG2, RESULTING IN A TRAP!
 3764  012134  005012                        CLR     (R2)
 3765  012136  000240                        NOP
 3766
 3767  012140                        MG3:                            ;REPORT ERROR.  MAINTENANCE
 3768  012140  010437  001230                MOV     R4,$TMP2        ;FUNCTION FAILED TO
 3769                                                                ;CAUSE ERROR.
 3770  012144  104127                1$:     ERROR   127
 3771  012146  012737  177777  030760  MOV   #-1,MANFL2
 3772  012154  000500                        BR      MGDONE
 3773
 3774  012156  022737  004410  177744  MGERRO: CMP     #4410,@#MEMERR  ;DID THE ERROR REGISTER
 3775  012164  001042                        BNE     69$             ;SET PROPERLY?
 3776
 3777  012166  022626                64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
 3778  012170  005037  177572        65$:    CLR     @#MMR0
 3779  012174  005037  172516                CLR     @#MMR3
 3780  012200  012737  177777  177744        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
 3781  012206  005737  177744                TST     @#MEMERR        ;REGISTER.
 3782  012212  001416                        BEQ     68$
 3783
 3784  012214                        66$:                            ;ERROR REGISTER WON'T
 3785  012214  013737  177740  001230        MOV     @#LOADRS,$TMP2  ;CLEAR
 3786  012222  013737  177742  001232        MOV     @#HIADRS,$TMP3
 3787  012230  013737  177744  001234        MOV     @#MEMERR,$TMP4
 3788
 3789  012236  104130                67$:    ERROR   130
 3790  012240  012737  177777  030740        MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
 3791  012246  000443                        BR      MGDONE
 3792
 3793  012250  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
 3794  012256  001356                        BNE     66$             ;UNLOCKED.
 3795  012260  022737  000003  177742        CMP     #3,@#HIADRS
 3796  012266  001352                        BNE     66$
 3797  012270  000432                        BR      MGDONE
 3798
 3799  012272                        69$:                            ;REPORT ERROR REGISTER
 3800  012272  012637  001230                MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
 3801  012276  005726                        TST     (SP)+           ;RESET THE STACK.
 3802  012300  013737  177740  001232        MOV     @#LOADRS,$TMP3
 3803  012306  013737  177742  001234        MOV     @#HIADRS,$TMP4
 3804  012314  012737  040000  001236        MOV     #40000,$TMP5
```

```
3805  012322  012737  004410  001240          MOV     #4410,$TMP6
3806  012330  013737  177744  001242          MOV     @#MEMERR,$TMP7
3807
3808  012336  104131                  70$:     ERROR   131
3809  012340  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3810  012346  012737  177777  030754          MOV     #-1,MMRFL2
3811  012354  000705                          BR      65$
3812  012356  104416              MGDONE:     RSET
3813
3814
3815                                ;;******************************************************************
3816                                ;*TEST 24       CACHE MAINTENANCE AND ERROR REGISTERS TEST 10
3817                                ;*
3818                                ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3819                                ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO, FOR THE
3820                                ;*LOW BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
3821                                ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3822                                ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3823                                ;*TO THE CACHE.
3824                                ;*
3825                                ;;******************************************************************
3826  012360  000004                TST24:    SCOPE
3827  012362  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3828            000024              MH=$TN-1
3829                                                                  ;SET THE SKAD REGISTER
3830  012370  012737  012724  030524          MOV     #TST25,SKAD     ;IN CASE THE TEST ABORTS.
3831
3832  012376  113737  001102  001224          MOVB    $TSTNM,$TMP0
3833
3834  012404  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3835  012406  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3836  012410  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3837  012412  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3838  012414  012737  012522  000114          MOV     #MHERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
3839  012422  012704  000400                  MOV     #400,R4 ;PATTERN TO BE PUT IN MAINT. REG.
3840  012426  012702  177750                  MOV     #MAINT,R2
3841  012432  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
3842                                                                  ;FORCE MISS THE OTHER
3843                                                                  ;GROUP
3844  012440  012705  012502                  MOV     #MH1,R5         ;MAKE MH1 A HIT IN
3845  012444  005715                          TST     (R5)            ;GROUP GP.
3846  012446  005715                          TST     (R5)
3847
3848                                                                  ;SEE IF REFERENCE ADDRESS
3849  012450  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
3850  012456  001007                          BNE     1$
3851                                                                  ;IF NOT ERROR!
3852  012460  010537  001230                  MOV     R5,$TMP2
3853  012464  012737  000000  001226          MOV     #0,$TMP1
3854  012472  104001                          ERROR   1
3855
3856  012474  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
3857
3858  012476  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
3859  012500  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
3860  012502  005012                  MH1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
```

```
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
                                                                ;CACHE ADDRESS MEMORY GROUP GP.


012504                          MH2:                            ;REPORT ERROR.  MAINTENANCE
012504  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
012510  104127                  1$:     ERROR   127
012512  012737  177777  030760  MOV     #-1,MANFL2
012520  000500                  BR      MHDONE

012522  022737  004420  177744  MHERRO: CMP     #4420,@#MEMERR  ;DID THE ERROR REGISTER
012530  001042                  BNE     69$                     ;SET PROPERLY?

012532  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
012534  005037  177572          65$:    CLR     @#MMR0
012540  005037  172516                  CLR     @#MMR3
012544  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
012552  005737  177744                  TST     @#MEMERR        ;REGISTER.
012556  001416                          BEQ     68$

012560                          66$:                            ;ERROR REGISTER WON'T
012560  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
012566  013737  177742  001232          MOV     @#HIADRS,$TMP3
012574  013737  177744  001234          MOV     @#MEMERR,$TMP4

012602  104130                  67$:    ERROR   130
012604  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
012612  000443                          BR      MHDONE

012614  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
012622  001356                          BNE     66$             ;UNLOCKED.
012624  022737  000003  177742          CMP     #3,@#HIADRS
012632  001352                          BNE     66$
012634  000432                          BR      MHDONE

012636                          69$:                            ;REPORT ERROR REGISTER
012636  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
012642  005726                          TST     (SP)+           ;RESET THE STACK.
012644  013737  177740  001232          MOV     @#LOADRS,$TMP3
012652  013737  177742  001234          MOV     @#HIADRS,$TMP4
012660  012737  000400  001236          MOV     #400,$TMP5
012666  012737  004420  001240          MOV     #4420,$TMP6
012674  013737  177744  001242          MOV     @#MEMERR,$TMP7

012702  104131                  70$:    ERROR   131
012704  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
012712  012737  177777  030754          MOV     #-1,MMRFL2
012720  000705                          BR      65$
012722  104416          MHDONE: RSET


                        ;;*****************************************************************
                        ;*TEST 25        CACHE MAINTENANCE AND ERROR REGISTERS TEST 11
                        ;*
```

```
                                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
                                        ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO, FOR THE
                                        ;*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
                                        ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
                                        ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
                                        ;*TO THE CACHE.
                                        ;*
                                        ;**************************************************************
012724  000004                TST25:  SCOPE
012726  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000025                  MI=$TN-1
                                                                ;SET THE SKAD REGISTER
012734  012737  013270  030524          MOV     #TST26,SKAD     ;IN CASE THE TEST ABORTS.

012742  113737  001102  001224          MOVB    $TSTNM,$TMP0

012750  104430                        SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
012752  104432                        SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
012754  104434                        SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
012756  104436                        SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
012760  012737  013066  000114        MOV     #MIERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
012766  012704  001000                MOV     #1000,R4        ;PATTERN TO BE PUT IN MAINT. REG.
012772  012702  177750                MOV     #MAINT,R2
012776  012737  000030  177746        MOV     #SOMI,@#CONTRL  ;FORCE SELECT GROUP 0 AND
                                                              ;FORCE MISS THE OTHER
                                                              ;GROUP
013004  012705  013046                MOV     #MI1,R5         ;MAKE MI1 A HIT IN
013010  005715                        TST     (R5)            ;GROUP GP.
013012  005715                        TST     (R5)

                                                              ;SEE IF REFERENCE ADDRESS
013014  032737  000010  177752        BIT     #10,@#HITMIS    ;IS A HIT.
013022  001007                        BNE     1$

                                                              ;IF NOT ERROR!
013024  010537  001230                MOV     R5,$TMP2
013030  012737  000000  001226        MOV     #0,$TMP1
013036  104001                        ERROR   1

013040  104420                        SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

013042  000240                1$:     NOP                     ;PUT THE PATTERN IN THE
013044  010412                        MOV     R4,(R2)         ;MAINTENANCE REGISTER.
013046  005012                MI1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                              ;INSTRUCTION SHOULD CAUSE
                                                              ;A PARITY ERROR IN THE
                                                              ;CACHE ADDRESS MEMORY GROUP GP.

013050                        MI2:                            ;REPORT ERROR.  MAINTENANCE
013050  010437  001230                MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                              ;CAUSE ERROR.
013054  104127                1$:     ERROR   127
013056  012737  177777  030760  MOV   #-1,MANFL2
013064  000500                        BR      MIDONE

013066  022737  004420  177744  MIERRO: CMP   #4420,@#MEMERR  ;DID THE ERROR REGISTER
```

```
3973  013074  001042                          BNE     69$             ;SET PROPERLY?
3974
3975  013076  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3976  013100  005037  177572          65$:    CLR     @#MMR0
3977  013104  005037  172516                  CLR     @#MMR3
3978  013110  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3979  013116  005737  177744                  TST     @#MEMERR        ;REGISTER.
3980  013122  001416                          BEQ     68$
3981
3982  013124                  66$:                                    ;ERROR REGISTER WON'T
3983  013124  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
3984  013132  013737  177742  001232          MOV     @#HIADRS,$TMP3
3985  013140  013737  177744  001234          MOV     @#MEMERR,$TMP4
3986
3987  013146  104130                  67$:    ERROR   130
3988  013150  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3989  013156  000443                          BR      MIDONE
3990
3991  013160  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3992  013166  001356                          BNE     66$             ;UNLOCKED.
3993  013170  022737  000003  177742          CMP     #3,@#HIADRS
3994  013176  001352                          BNE     66$
3995  013200  000432                          BR      MIDONE
3996
3997  013202                  69$:                                    ;REPORT ERROR REGISTER
3998  013202  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3999  013206  005726                          TST     (SP)+           ;RESET THE STACK.
4000  013210  013737  177740  001232          MOV     @#LOADRS,$TMP3
4001  013216  013737  177742  001234          MOV     @#HIADRS,$TMP4
4002  013224  012737  001000  001236          MOV     #1000,$TMP5
4003  013232  012737  004420  001240          MOV     #4420,$TMP6
4004  013240  013737  177744  001242          MOV     @#MEMERR,$TMP7
4005
4006  013246  104131                  70$:    ERROR   131
4007  013250  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
4008  013256  012737  177777  030754          MOV     #-1,MMRFL2
4009  013264  000705                          BR      65$
4010  013266  104416          MIDONE: RSET
4011
4012
4013                                  ;;**********************************************************************
4014                                  ;*TEST 26       CACHE MAINTENANCE AND ERROR REGISTERS TEST 12
4015                                  ;*
4016                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
4017                                  ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE
4018                                  ;*LOW BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
4019                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
4020                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
4021                                  ;*TO THE CACHE.
4022                                  ;*
4023                                  ;;**********************************************************************
4024  013270  000004          TST26:  SCOPE
4025  013272  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4026          000026          MJ=$TN-1
4027                                                                  ;SET THE SKAD REGISTER
4028  013300  012737  013634  030524          MOV     #TST27,SKAD     ;IN CASE THE TEST ABORTS.
```

```
4029
4030  013306  113737  001102  001224          MOVB    $TSTNM,$TMPO
4031
4032  013314  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4033  013316  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4034  013320  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4035  013322  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4036  013324  012737  013432  000114          MOV     #MJERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
4037  013332  012704  002000                  MOV     #2000,R4        ;PATTERN TO BE PUT IN MAINT. REG.
4038  013336  012702  177750                  MOV     #MAINT,R2
4039  013342  012737  000044  177746          MOV     #S1MO,@#CONTRL  ;FORCE SELECT GROUP 1 AND
4040                                                                  ;FORCE MISS THE OTHER
4041                                                                  ;GROUP
4042  013350  012705  013412                  MOV     #MJ1,R5         ;MAKE MJ1 A HIT IN
4043  013354  005715                          TST     (R5)            ;GROUP GP.
4044  013356  005715                          TST     (R5)
4045
4046                                                                  ;SEE IF REFERENCE ADDRESS
4047  013360  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
4048  013366  001007                          BNE     1$
4049                                                                  ;IF NOT ERROR!
4050  013370  010537  001230                  MOV     R5,$TMP2
4051  013374  012737  000001  001226          MOV     #1,$TMP1
4052  013402  104001                          ERROR   1
4053
4054  013404  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
4055
4056  013406  000240          1$:     NOP                             ;PUT THE PATTERN IN THE
4057  013410  010412                  MOV     R4,(R2)                 ;MAINTENANCE REGISTER.
4058  013412  005012          MJ1:    CLR     (R2)                    ;THE FETCH OF THIS NEXT
4059                                                                  ;INSTRUCTION SHOULD CAUSE
4060                                                                  ;A PARITY ERROR IN THE
4061                                                                  ;CACHE ADDRESS MEMORY GROUP GP.
4062
4063
4064  013414                  MJ2:                                    ;REPORT ERROR.  MAINTENANCE
4065  013414  010437  001230          MOV     R4,$TMP2                ;FUNCTION FAILED TO
4066                                                                  ;CAUSE ERROR.
4067  013420  104127          1$:     ERROR   127
4068  013422  012737  177777  030760  MOV     #-1,MANFL2
4069  013430  000500                  BR      MJDONE
4070
4071  013432  022737  004440  177744  MJERRO: CMP     #4440,@#MEMERR  ;DID THE ERROR REGISTER
4072  013440  001042                  BNE     69$             ;SET PROPERLY?
4073
4074  013442  022626          64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
4075  013444  005037  177572  65$:    CLR     @#MMR0
4076  013450  005037  172516          CLR     @#MMR3
4077  013454  012737  177777  177744  MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
4078  013462  005737  177744          TST     @#MEMERR        ;REGISTER.
4079  013466  001416                  BEQ     68$
4080
4081  013470                  66$:                            ;ERROR REGISTER WON'T
4082  013470  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
4083  013476  013737  177742  001232          MOV     @#HIADRS,$TMP3
4084  013504  013737  177744  001234          MOV     @#MEMERR,$TMP4
```

```

013512	104130				67$:	ERROR	130
013514	012737	177777	030740			MOV	#-1,MMRFLG		;SIGNAL BAD REGISTER
013522	000443					BR	MJDONE

013524	022737	177740	177740	68$:	CMP	#177740,@#LOADRS	;SEE IF ADDRESS REGISTER
013532	001356					BNE	66$			;UNLOCKED.
013534	022737	000003	177742			CMP	#3,@#HIADRS
013542	001352					BNE	66$
013544	000432					BR	MJDONE

013546						69$:				;REPORT ERROR REGISTER
013546	012637	001230				MOV	(SP)+,$TMP2		;NOT SET AS EXPECTED.
013552	005726					TST	(SP)+			;RESET THE STACK.
013554	013737	177740	001232			MOV	@#LOADRS,$TMP3
013562	013737	177742	001234			MOV	@#HIADRS,$TMP4
013570	012737	002000	001236			MOV	#2000,$TMP5
013576	012737	004440	001240			MOV	#4440,$TMP6
013604	013737	177744	001242			MOV	@#MEMERR,$TMP7

013612	104131				70$:	ERROR	131
013614	012737	177777	030760			MOV	#-1,MANFL2		;SIGNAL BAD REGISTER
013622	012737	177777	030754			MOV	#-1,MMRFL2
013630	000705					BR	65$
013632	104416				MJDONE: RSET



					;;*****************************************************************
					;*TEST 27		CACHE MAINTENANCE AND ERROR REGISTERS TEST 13
					;*
					;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
					;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE
					;*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
					;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
					;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
					;*TO THE CACHE.
					;*
					;;*****************************************************************
013634	000004				TST27:	SCOPE
013636	012737	000040	001274			MOV	#40,$TIMES		;;DO 40 ITERATIONS
	000027				MK=$TN-1
										;SET THE SKAD REGISTER
013644	012737	014200	030524			MOV	#TST30,SKAD		;IN CASE THE TEST ABORTS.

013652	113737	001102	001224			MOVB	$TSTNM,$TMP0

013660	104430					SKPBER			;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
013662	104432					SKPBCN			;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
013664	104434					SKPBMN			;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
013666	104436					SKPBHM			;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
013670	012737	013776	000114			MOV	#MKERR0,@#CACHVEC	;SET UP FOR THE ERROR.
013676	012704	004000				MOV	#4000,R4		;PATTERN TO BE PUT IN MAINT. REG.
013702	012702	177750				MOV	#MAINT,R2
013706	012737	000044	177746			MOV	#S1M0,@#CONTRL		;FORCE SELECT GROUP 1 AND
										;FORCE MISS THE OTHER
										;GROUP
```

```
                                        MOV     #MK1,R5         ;MAKE MK1 A HIT IN
                                        TST     (R5)            ;GROUP GP.
                                        TST     (R5)
                                                                ;SEE IF REFERENCE ADDRESS
                                        BIT     #10,@#HITMIS    ;IS A HIT.
                                        BNE     1$
                                                                ;IF NOT ERROR!
                                        MOV     R5,$TMP2
                                        MOV     #1,$TMP1
                                        ERROR   1
                                        SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
                                1$:     NOP                     ;PUT THE PATTERN IN THE
                                        MOV     R4,(R2)         ;MAINTENANCE REGISTER.
                                MK1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
                                                                ;CACHE ADDRESS MEMORY GROUP GP.
                                MK2:                            ;REPORT ERROR.  MAINTENANCE
                                        MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
                                1$:     ERROR   127
                                MOV     #-1,MANFL2
                                        BR      MKDONE
                        MKERR0: CMP     #4440,@#MEMERR  ;DID THE ERROR REGISTER
                                        BNE     69$             ;SET PROPERLY?
                                64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
                                65$:    CLR     @#MMR0
                                        CLR     @#MMR3
                                        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
                                        TST     @#MEMERR        ;REGISTER.
                                        BEQ     68$
                                66$:                            ;ERROR REGISTER WON'T
                                        MOV     @#LOADRS,$TMP2  ;CLEAR
                                        MOV     @#HIADRS,$TMP3
                                        MOV     @#MEMERR,$TMP4
                                67$:    ERROR   130
                                        MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
                                        BR      MKDONE
                        68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
                                        BNE     66$             ;UNLOCKED.
                                        CMP     #3,@#HIADRS
                                        BNE     66$
                                        BR      MKDONE
                                69$:                            ;REPORT ERROR REGISTER
                                        MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
```

```
                                        TST     (SP)+           ;RESET THE STACK.
                                        MOV     @#LOADRS,$TMP3
                                        MOV     @#HIADRS,$TMP4
                                        MOV     #4000,$TMP5
                                        MOV     #4440,$TMP6
                                        MOV     @#MEMERR,$TMP7

                                70$:    ERROR   131
                                        MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
                                        MOV     #-1,MMRFL2
                                        BR      65$
                                MKDONE: RSET


                                ;;*****************************************************************
                                ;*TEST 30        CACHE MAINTENANCE AND ERROR REGISTERS TEST 14
                                ;*
                                ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
                                ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ZERO, FOR THE
                                ;*LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
                                ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
                                ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
                                ;*TO THE CACHE.
                                ;*
                                ;;*****************************************************************
                                TST30:  SCOPE
                                        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                ML=$TN-1
                                                                ;SET THE SKAD REGISTER
                                        MOV     #TST31,SKAD     ;IN CASE THE TEST ABORTS.

                                        MOVB    $TSTNM,$TMP0

                                        SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
                                        SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
                                        SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
                                        SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
                                        MOV     #MLERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
                                        MOV     #20,R4  ;PATTERN TO BE PUT IN MAINT. REG.
                                        MOV     #MAINT,R2
                                        MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
                                                                ;FORCE MISS THE OTHER
                                                                ;GROUP
                                        MOV     #ML1,R5         ;MAKE ML1 A HIT IN
                                        TST     (R5)            ;GROUP GP.
                                        TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
                                        BIT     #10,@#HITMIS    ;IS A HIT.
                                        BNE     1$
                                                                ;IF NOT ERROR!
                                        MOV     R5,$TMP2
                                        MOV     #0,$TMP1
                                        ERROR   1

                                        SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
```

```
4253
4254  014316  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
4255  014320  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
4256  014322  005012                  ML1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
4257                                                                  ;INSTRUCTION SHOULD CAUSE
4258                                                                  ;A PARITY ERROR IN THE
4259                                                                  ;CACHE DATA MEMORY GROUP GP.
4260
4261
4262  014324                          ML2:                            ;REPORT ERROR.  MAINTENANCE
4263  014324  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
4264                                                                  ;CAUSE ERROR.
4265  014330  104127                  1$:     ERROR   127
4266  014332  012737  177777  030760  MOV     #-1,MANFL2
4267  014340  000500                          BR      MLDONE
4268
4269  014342  022737  004500  177744  MLERR0: CMP     #4500,@#MEMERR  ;DID THE ERROR REGISTER
4270  014350  001042                          BNE     69$             ;SET PROPERLY?
4271
4272  014352  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
4273  014354  005037  177572          65$:    CLR     @#MMR0
4274  014360  005037  172516                  CLR     @#MMR3
4275  014364  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
4276  014372  005737  177744                  TST     @#MEMERR        ;REGISTER.
4277  014376  001416                          BEQ     68$
4278
4279  014400                          66$:                            ;ERROR REGISTER WON'T
4280  014400  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
4281  014406  013737  177742  001232          MOV     @#HIADRS,$TMP3
4282  014414  013737  177744  001234          MOV     @#MEMERR,$TMP4
4283
4284  014422  104130                  67$:    ERROR   130
4285  014424  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
4286  014432  000443                          BR      MLDONE
4287
4288  014434  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
4289  014442  001356                          BNE     66$             ;UNLOCKED.
4290  014444  022737  000003  177742          CMP     #3,@#HIADRS
4291  014452  001352                          BNE     66$
4292  014454  000432                          BR      MLDONE
4293
4294  014456                          69$:                            ;REPORT ERROR REGISTER
4295  014456  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
4296  014462  005726                          TST     (SP)+           ;RESET THE STACK.
4297  014464  013737  177740  001232          MOV     @#LOADRS,$TMP3
4298  014472  013737  177742  001234          MOV     @#HIADRS,$TMP4
4299  014500  012737  000020  001236          MOV     #20,$TMP5
4300  014506  012737  004500  001240          MOV     #4500,$TMP6
4301  014514  013737  177744  001242          MOV     @#MEMERR,$TMP7
4302
4303  014522  104131                  70$:    ERROR   131
4304  014524  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
4305  014532  012737  177777  030754          MOV     #-1,MMRFL2
4306  014540  000705                          BR      65$
4307  014542  104416          MLDONE: RSET
4308
```

```
                                        ;;*****************************************************************
                                        ;*TEST 31       CACHE MAINTENANCE AND ERROR REGISTERS TEST 15
                                        ;*
                                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
                                        ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ZERO, FOR THE
                                        ;*HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
                                        ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
                                        ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
                                        ;*TO THE CACHE.
                                        ;*
                                        ;*****************************************************************
014544  000004                          TST31:  SCOPE
014546  012737  000040  001274                  MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000031                          MN=$TN-1
                                                                        ;SET THE SKAD REGISTER
014554  012737  015110  030524                  MOV     #TST32,SKAD     ;IN CASE THE TEST ABORTS.

014562  113737  001102  001224                  MOVB    $TSTNM,$TMPO

014570  104430                                  SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
014572  104432                                  SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
014574  104434                                  SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
014576  104436                                  SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
014600  012737  014706  000114                  MOV     #NMERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
014606  012704  000040                          MOV     #40,R4  ;PATTERN TO BE PUT IN MAINT. REG.
014612  012702  177750                          MOV     #MAINT,R2
014616  012737  000030  177746                  MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
                                                                        ;FORCE MISS THE OTHER
                                                                        ;GROUP
014624  012705  014666                          MOV     #NM1,R5         ;MAKE NM1 A HIT IN
014630  005715                                  TST     (R5)            ;GROUP GP.
014632  005715                                  TST     (R5)

                                                                        ;SEE IF REFERENCE ADDRESS
014634  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
014642  001007                                  BNE     1$

                                                                        ;IF NOT ERROR!
014644  010537  001230                          MOV     R5,$TMP2
014650  012737  000000  001226                  MOV     #0,$TMP1
014656  104001                                  ERROR   1

014660  104420                                  SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

014662  000240                          1$:     NOP                     ;PUT THE PATTERN IN THE
014664  010412                                  MOV     R4,(R2)         ;MAINTENANCE REGISTER.
014666  005012                          NM1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                        ;INSTRUCTION SHOULD CAUSE
                                                                        ;A PARITY ERROR IN THE
                                                                        ;CACHE DATA MEMORY GROUP GP.

014670                                  NM2:                            ;REPORT ERROR.  MAINTENANCE
014670  010437  001230                          MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                        ;CAUSE ERROR.
014674  104127                          1$:     ERROR   127
```

```
 4365  014676  012737  177777  030760  MOV     #-1,MANFL2
 4366  014704  000500                  BR      NMDONE
 4367
 4368  014706  022737  004500  177744  NMERR0: CMP     #4500,@#MEMERR  ;DID THE ERROR REGISTER
 4369  014714  001042                  BNE     69$             ;SET PROPERLY?
 4370
 4371  014716  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
 4372  014720  005037  177572          65$:    CLR     @#MMR0
 4373  014724  005037  172516          CLR     @#MMR3
 4374  014730  012737  177777  177744  MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
 4375  014736  005737  177744          TST     @#MEMERR        ;REGISTER.
 4376  014742  001416                  BEQ     68$
 4377
 4378  014744                          66$:                    ;ERROR REGISTER WON'T
 4379  014744  013737  177740  001230  MOV     @#LOADRS,$TMP2  ;CLEAR
 4380  014752  013737  177742  001232  MOV     @#HIADRS,$TMP3
 4381  014760  013737  177744  001234  MOV     @#MEMERR,$TMP4
 4382
 4383  014766  104130                  67$:    ERROR   130
 4384  014770  012737  177777  030740  MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
 4385  014776  000443                  BR      NMDONE
 4386
 4387  015000  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
 4388  015006  001356                  BNE     66$             ;UNLOCKED.
 4389  015010  022737  000003  177742  CMP     #3,@#HIADRS
 4390  015016  001352                  BNE     66$
 4391  015020  000432                  BR      NMDONE
 4392
 4393  015022                          69$:                    ;REPORT ERROR REGISTER
 4394  015022  012637  001230          MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
 4395  015026  005726                  TST     (SP)+           ;RESET THE STACK.
 4396  015030  013737  177740  001232  MOV     @#LOADRS,$TMP3
 4397  015036  013737  177742  001234  MOV     @#HIADRS,$TMP4
 4398  015044  012737  000040  001236  MOV     #40,$TMP5
 4399  015052  012737  004500  001240  MOV     #4500,$TMP6
 4400  015060  013737  177744  001242  MOV     @#MEMERR,$TMP7
 4401
 4402  015066  104131                  70$:    ERROR   131
 4403  015070  012737  177777  030760  MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
 4404  015076  012737  177777  030754  MOV     #-1,MMRFL2
 4405  015104  000705                  BR      65$
 4406  015106  104416                  NMDONE: RSET
 4407
 4408
 4409                                  ;;*****************************************************************
 4410                                  ;*TEST 32        CACHE MAINTENANCE AND ERROR REGISTERS TEST 16
 4411                                  ;*
 4412                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
 4413                                  ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
 4414                                  ;*LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
 4415                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
 4416                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
 4417                                  ;*TO THE CACHE.
 4418                                  ;*
 4419                                  ;;*****************************************************************
 4420  015110  000004                  TST32:  SCOPE
```

```
4421  015112  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4422          000032                  M0=$TN-1
4423                                                  ;SET THE SKAD REGISTER
4424  015120  012737  015454  030524          MOV     #TST33,SKAD     ;IN CASE THE TEST ABORTS.
4425
4426  015126  113737  001102  001224          MOVB    $TSTNM,$TMP0
4427
4428  015134  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4429  015136  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4430  015140  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4431  015142  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4432  015144  012737  015252  000114          MOV     #MOERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
4433  015152  012704  000100                  MOV     #100,R4 ;PATTERN TO BE PUT IN MAINT. REG.
4434  015156  012702  177750                  MOV     #MAINT,R2
4435  015162  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
4436                                                                  ;FORCE MISS THE OTHER
4437                                                                  ;GROUP
4438  015170  012705  015232                  MOV     #M01,R5         ;MAKE M01 A HIT IN
4439  015174  005715                          TST     (R5)            ;GROUP GP.
4440  015176  005715                          TST     (R5)
4441
4442                                                                  ;SEE IF REFERENCE ADDRESS
4443  015200  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
4444  015206  001007                          BNE     1$
4445                                                                  ;IF NOT ERROR!
4446  015210  010537  001230                  MOV     R5,$TMP2
4447  015214  012737  000001  001226          MOV     #1,$TMP1
4448  015222  104001                          ERROR   1
4449
4450  015224  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
4451
4452  015226  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
4453  015230  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
4454  015232  005012                  M01:    CLR     (R2)            ;THE FETCH OF THIS NEXT
4455                                                                  ;INSTRUCTION SHOULD CAUSE
4456                                                                  ;A PARITY ERROR IN THE
4457                                                                  ;CACHE DATA MEMORY GROUP GP.
4458
4459
4460  015234                          M02:                            ;REPORT ERROR.  MAINTENANCE
4461  015234  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
4462                                                                  ;CAUSE ERROR.
4463  015240  104127                  1$:     ERROR   127
4464  015242  012737  177777  030760  MOV     #-1,MANFL2
4465  015250  000500                          BR      MODONE
4466
4467  015252  022737  004600  177744  MOERRO: CMP     #4600,@#MEMERR  ;DID THE ERROR REGISTER
4468  015260  001042                          BNE     69$             ;SET PROPERLY?
4469
4470  015262  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
4471  015264  005037  177572          65$:    CLR     @#MMR0
4472  015270  005037  172516                  CLR     @#MMR3
4473  015274  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
4474  015302  005737  177744                  TST     @#MEMERR        ;REGISTER.
4475  015306  001416                          BEQ     68$
4476
```

```
 015310                             66$:                             ;ERROR REGISTER WON'T
 015310  013737  177740  001230             MOV     @#LOADRS,$TMP2   ;CLEAR
 015316  013737  177742  001232             MOV     @#HIADRS,$TMP3
 015324  013737  177744  001234             MOV     @#MEMERR,$TMP4

 015332  104130                     67$:    ERROR   130
 015334  012737  177777  030740             MOV     #-1,MMRFLG       ;SIGNAL BAD REGISTER
 015342  000443                             BR      MODONE

 015344  022737  177740  177740     68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
 015352  001356                             BNE     66$              ;UNLOCKED.
 015354  022737  000003  177742             CMP     #3,@#HIADRS
 015362  001352                             BNE     66$
 015364  000432                             BR      MODONE

 015366                             69$:                             ;REPORT ERROR REGISTER
 015366  012637  001230                     MOV     (SP)+,$TMP2      ;NOT SET AS EXPECTED.
 015372  005726                             TST     (SP)+            ;RESET THE STACK.
 015374  013737  177740  001232             MOV     @#LOADRS,$TMP3
 015402  013737  177742  001234             MOV     @#HIADRS,$TMP4
 015410  012737  000100  001236             MOV     #100,$TMP5
 015416  012737  004600  001240             MOV     #4600,$TMP6
 015424  013737  177744  001242             MOV     @#MEMERR,$TMP7

 015432  104131                     70$:    ERROR   131
 015434  012737  177777  030760             MOV     #-1,MANFL2       ;SIGNAL BAD REGISTER
 015442  012737  177777  030754             MOV     #-1,MMRFL2
 015450  000705                             BR      65$
 015452  104416                     MODONE: RSET


                                    ;;*************************************************************
                                    ;*TEST 33        CACHE MAINTENANCE AND ERROR REGISTERS TEST 17
                                    ;*
                                    ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
                                    ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
                                    ;*HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
                                    ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
                                    ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
                                    ;*TO THE CACHE.
                                    ;*
                                    ;;*************************************************************
 015454  000004                     TST33:  SCOPE
 015456  012737  000040  001274             MOV     #40,$TIMES       ;;DO 40 ITERATIONS
         000033                     MP=$TN-1
                                                                     ;SET THE SKAD REGISTER
 015464  012737  016020  030524             MOV     #TST34,SKAD      ;IN CASE THE TEST ABORTS.

 015472  113737  001102  001224             MOVB    $TSTNM,$TMP0

 015500  104430                             SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
 015502  104432                             SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
 015504  104434                             SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
 015506  104436                             SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
 015510  012737  015616  000114             MOV     #MPERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
 015516  012704  000200                     MOV     #200,R4 ;PATTERN TO BE PUT IN MAINT. REG.
```

```
4533  015522  012702  177750                MOV     #MAINT,R2
4534  015526  012737  000044  177746        MOV     #S1MO,@#CONTRL  ;FORCE SELECT GROUP 1 AND
4535                                                                ;FORCE MISS THE OTHER
4536                                                                ;GROUP
4537  015534  012705  015576                MOV     #MP1,R5         ;MAKE MP1 A HIT IN
4538  015540  005715                        TST     (R5)            ;GROUP GP.
4539  015542  005715                        TST     (R5)
4540
4541                                                                ;SEE IF REFERENCE ADDRESS
4542  015544  032737  000010  177752        BIT     #10,@#HITMIS    ;IS A HIT.
4543  015552  001007                        BNE     1$
4544                                                                ;IF NOT ERROR!
4545  015554  010537  001230                MOV     R5,$TMP2
4546  015560  012737  000001  001226        MOV     #1,$TMP1
4547  015566  104001                        ERROR   1
4548
4549  015570  104420                        SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
4550
4551  015572  000240              1$:       NOP                     ;PUT THE PATTERN IN THE
4552  015574  010412                        MOV     R4,(R2)         ;MAINTENANCE REGISTER.
4553  015576  005012              MP1:      CLR     (R2)            ;THE FETCH OF THIS NEXT
4554                                                                ;INSTRUCTION SHOULD CAUSE
4555                                                                ;A PARITY ERROR IN THE
4556                                                                ;CACHE DATA MEMORY GROUP GP.
4557
4558
4559  015600              MP2:                                      ;REPORT ERROR.  MAINTENANCE
4560  015600  010437  001230                MOV     R4,$TMP2        ;FUNCTION FAILED TO
4561                                                                ;CAUSE ERROR.
4562  015604  104127              1$:       ERROR   127
4563  015606  012737  177777  030760  MOV   #-1,MANFL2
4564  015614  000500                        BR      MPDONE
4565
4566  015616  022737  004600  177744  MPERRO: CMP     #4600,@#MEMERR  ;DID THE ERROR REGISTER
4567  015624  001042                        BNE     69$             ;SET PROPERLY?
4568
4569  015626  022626              64$:      CMP     (SP)+,(SP)+     ;RESET THE STACK
4570  015630  005037  177572      65$:      CLR     @#MMR0
4571  015634  005037  172516              CLR     @#MMR3
4572  015640  012737  177777  177744        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
4573  015646  005737  177744                TST     @#MEMERR        ;REGISTER.
4574  015652  001416                        BEQ     68$
4575
4576  015654              66$:                                      ;ERROR REGISTER WON'T
4577  015654  013737  177740  001230        MOV     @#LOADRS,$TMP2  ;CLEAR
4578  015662  013737  177742  001232        MOV     @#HIADRS,$TMP3
4579  015670  013737  177744  001234        MOV     @#MEMERR,$TMP4
4580
4581  015676  104130              67$:      ERROR   130
4582  015700  012737  177777  030740        MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
4583  015706  000443                        BR      MPDONE
4584
4585  015710  022737  177740  177740  68$:      CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
4586  015716  001356                        BNE     66$             ;UNLOCKED.
4587  015720  022737  000003  177742        CMP     #3,@#HIADRS
4588  015726  001352                        BNE     66$
```

```
4589  015730  000432                          BR      MPDONE
4590
4591  015732                          69$:                            ;REPORT ERROR REGISTER
4592  015732  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
4593  015736  005726                          TST     (SP)+           ;RESET THE STACK.
4594  015740  013737  177740  001232          MOV     @#LOADRS,$TMP3
4595  015746  013737  177742  001234          MOV     @#HIADRS,$TMP4
4596  015754  012737  000200  001236          MOV     #200,$TMP5
4597  015762  012737  004600  001240          MOV     #4600,$TMP6
4598  015770  013737  177744  001242          MOV     @#MEMERR,$TMP7
4599
4600  015776  104131                  70$:    ERROR   131
4601  016000  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
4602  016006  012737  177777  030754          MOV     #-1,MMRFL2
4603  016014  000705                          BR      65$
4604  016016  104416                  MPDONE: RSET
4605
4606
4607
4608
4609
4610                                  ;;*****************************************************************
4611                                  ;*TEST 34       CACHE MAINTENANCE AND ERROR REGISTERS TEST 20
4612                                  ;*
4613                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
4614                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
4615                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
4616                                  ;*THE MAINTENANCE REGISTER IS USED TO MAKE THAT REFERENCE CAUSE A
4617                                  ;*MAIN MEMORY ADDRESS AND CONTROL LINES PARITY ERROR ON THE
4618                                  ;*MAIN MEMORY BUS.
4619                                  ;*
4620                                  ;*****************************************************************
4621  016020  000004                  TST34:  SCOPE
4622  016022  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4623          000034                  MR=$TN-1
4624                                                                  ;SET THE SKAD REGISTER
4625  016030  012737  016450  030524          MOV     #TST35,SKAD     ;IN CASE THE TEST ABORTS.
4626
4627  016036  113737  001102  001224          MOVB    $TSTNM,$TMP0
4628
4629  016044  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4630  016046  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4631  016050  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4632  016052  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4633  016054  104422                          MMSKIP
4634  016056  012737  016240  000114          MOV     #MRERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
4635  016064  012737  030352  000004          MOV     #CPSPUR,@#ERRVEC        ;NOTE THAT WHEN THIS ERROR
4636                                                                  ;ON THE MAIN MEMORY ADDRESS
4637                                                                  ;AND CONTROL LINES OCCURS
4638                                                                  ;A TIME OUT WILL RESULT ON THE
4639                                                                  ;UNIBUS!! THIS WILL CAUSE A
4640                                                                  ;TRAP TO VECTOR ERRVEC BEFORE
4641                                                                  ;THE TRAP TO CACHVEC OCCURS! BOTH
4642                                                                  ;WILL OCCUR!
4643  016072  012746  177777                  MOV     #-1,-(SP)       ;PUT A MARKER ON THE STACK
4644
```

```
4645  016076  012700  172340              MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
4646                                                              ;TO RELOCATE EVERYTHING
4647  016102  012702  172300              MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
4648  016106  012703  000007              MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
4649  016112  005004                      CLR     R4              ;BY PASSIVELY IS MEANT
4650  016114  012705  170200              MOV     #MAPLOD,R5      ;THAT ADDRESS ARE
4651                                                              ;RELOCATED TO THEMSELVES.
4652  016120  012722  077406      64$:    MOV     #77406,(R2)+
4653  016124  010401                      MOV     R4,R1
4654  016126  072127  000006              ASH     #6,R1
4655  016132  010125                      MOV     R1,(R5)+
4656  016134  005025                      CLR     (R5)+
4657  016136  010410                      MOV     R4,(R0)
4658  016140  062720  170000              ADD     #170000,(R0)+
4659  016144  062704  000200              ADD     #200,R4
4660  016150  077315                      SOB     R3,64$
4661  016152  012710  177600              MOV     #177600,(R0)
4662  016156  012712  077406              MOV     #77406,(R2)
4663
4664  016162  012737  000060  172516      MOV     #60,@#MMR3      ;TURN ON THE MAPPING BOX AND
4665  016170  012737  000001  177572      MOV     #1,@#MMR0       ;ENABLE 22 BIT MODE ADDRESSING.
4666
4667  016176  012737  000014  177746      MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
4668  016204  012702  177750              MOV     #MAINT,R2
4669  016210  000240                      NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
4670  016212  012712  000002              MOV     #2,(R2)         ;SET UP THE FORCE ERROR BIT IN
4671                                                              ;THE MAINTENANCE REGISTER.
4672  016216  005012                      CLR     (R2)            ;THE FETCH OF THIS INSTRUCTION
4673                                                              ;SHOULD RESULT IN A PARITY ERROR
4674                                                              ;ON THE MAIN MEMORY ADDRESS AND CONTROL
4675                                                              ;LINES. BECAUSE THIS REFERENCE
4676                                                              ;IS BEING MADE OVER THE UNIBUS
4677                                                              ;A UNIBUS TIME OUT WILL OCCUR
4678                                                              ;RESULTING IN AN ABORT TO VECTOR
4679                                                              ;ERRVEC. THEN IMMEDIATELY FOLLOWING
4680                                                              ;THIS ABORT TO ERRVEC, THE
4681                                                              ;PARITY ERROR WILL CAUSE A TRAP
4682                                                              ;TO CACHVEC!!!
4683
4684  016220                      MR1:                            ;REPORT FAILURE OF THE MAINTENANCE
4685  016220  012737  000002  001230      MOV     #2,$TMP2        ;TO FORCE THE ERROR.
4686  016226  104127              1$:     ERROR   127
4687  016230  012737  177777  030760      MOV     #-1,MANFL2
4688  016236  000503                      BR      MRDONE
4689
4690  016240  022766  177777  000010  MRERR0: CMP     #-1,10(SP)      ;DID 2 TRAPS OCCUR? SEE WHERE
4691                                                              ;THE MARKER IS ON THE STACK!
4692  016246  001401                      BEQ     MR2
4693  016250  104000                      ERROR
4694
4695  016252  022737  002402  177744  MR2:    CMP     #2402,@#MEMERR  ;DID THE ERROR REGISTER GET
4696  016260  001430                      BEQ     MR3             ;SET CORRECTLY.
4697
4698                                                              ;IF NOT REPORT THE ERROR.
4699  016262  022626                      CMP     (SP)+,(SP)+
4700  016264  012637  001230              MOV     (SP)+,$TMP2
```

```
4701  016270  022626                        CMP     (SP)+,(SP)+
4702  016272  013737  177740  001232        MOV     @#LOADRS,$TMP3
4703  016300  013737  177742  001234        MOV     @#HIADRS,$TMP4
4704  016306  012737  000002  001236        MOV     #2,$TMP5
4705  016314  012737  002402  001240        MOV     #2402,$TMP6
4706  016322  013737  177744  001242        MOV     @#MEMERR,$TMP7
4707  016330  104131                1$:     ERROR   131
4708  016332  012737  177777  030760        MOV     #-1,MANFL2
4709  016340  000402                        BR      MR4
4710
4711  016342  062706  000012        MR3:    ADD     #12,SP                  ;RESET THE STACK.
4712
4713  016346  005037  177572        MR4:    CLR     @#MMR0
4714  016352  005037  172516                CLR     @#MMR3
4715  016356  012737  177777  177744        MOV     #-1,@#MEMERR            ;TRY TO CLR THE ERROR REG.
4716  016364  005737  177744                TST     @#MEMERR
4717  016370  001416                        BEQ     MR6
4718
4719  016372                        MR5:                                    ;THE ERROR REGISTER WON'T CLR.
4720  016372  013737  177740  001230        MOV     @#LOADRS,$TMP2
4721  016400  013737  177742  001232        MOV     @#HIADRS,$TMP3
4722  016406  013737  177744  001234        MOV     @#MEMERR,$TMP4
4723  016414  104130                1$:     ERROR   130
4724  016416  012737  177777  030740        MOV     #-1,MMRFLG
4725  016424  000410                        BR      MRDONE
4726
4727  016426  022737  177740  177740  MR6:  CMP     #177740,@#LOADRS        ;SEE IF THE ADDRESS REGISTER
4728  016434  001356                        BNE     MR5                     ;GOT RESET.
4729  016436  022737  000003  177742        CMP     #3,@#HIADRS
4730  016444  001352                        BNE     MR5
4731
4732  016446  104416              MRDONE: RSET
4733
4734                                ;;**********************************************************************
4735                                ;*TEST 35       CACHE MAINTENANCE AND ERROR REGISTERS TEST 21
4736                                ;*
4737                                ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
4738                                ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
4739                                ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
4740                                ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
4741                                ;*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN EVEN WORD IN THE
4742                                ;*PAIR, WHICH IS ALSO THE WANTED WORD.
4743                                ;*
4744                                ;;**********************************************************************
4745  016450  000004              TST35:  SCOPE
4746  016452  012737  000040  001274        MOV     #40,$TIMES              ;;DO 40 ITERATIONS
4747          000035              MS=$TN-1
4748                                                                        ;SET THE SKAD REGISTER
4749  016460  012737  017070  030524        MOV     #TST36,SKAD             ;IN CASE THE TEST ABORTS.
4750
4751  016466  113737  001102  001224        MOVB    $TSTNM,$TMP0
4752
4753  016474  104430                        SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4754  016476  104432                        SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4755  016500  104434                        SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4756  016502  104436                        SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
```

```
                                        MMSKIP
016504  104422
016506  012737  016666  000114        MOV     #MSERR0,@#CACHVEC       ;SET UP FOR THE ERROR
016514  012700  172340                MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                              ;TO RELOCATE EVERYTHING
016520  012702  172300                MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
016524  012703  000007                MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
016530  005004                        CLR     R4              ;BY PASSIVELY IS MEANT
016532  012705  170200                MOV     #MAPLOO,R5      ;THAT ADDRESS ARE
                                                              ;RELOCATED TO THEMSELVES.
016536  012722  077406        64$:    MOV     #77406,(R2)+
016542  010401                        MOV     R4,R1
016544  072127  000006                ASH     #6,R1
016550  010125                        MOV     R1,(R5)+
016552  005025                        CLR     (R5)+
016554  010410                        MOV     R4,(R0)
016556  062720  170000                ADD     #170000,(R0)+
016562  062704  000200                ADD     #200,R4
016566  077315                        SOB     R3,64$
016570  012710  177600                MOV     #177600,(R0)
016574  012712  077406                MOV     #77406,(R2)

016600  012737  000060  172516        MOV     #60,@#MMR3              ;TURN THE MAP AND ENABLE
016606  012737  000001  177572        MOV     #1,@#MMR0               ;22 BIT MODE ADDRESSING.
016614  012704  010000                MOV     #10000,R4               ;PATTERN FOR THE MAINTENANCE
016620  012702  177750                MOV     #MAINT,R2               ;REGISTER.
016624  012737  000014  177746        MOV     #M1M0,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
016632  000402                        BR      MS1

        016634                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        016634                          LOC=-4&LOC
        016640                          LOC=LOC+4
        016640                          .=LOC

016640  000240          MS1:    NOP
016642  010412                  MOV     R4,(R2)                 ;TURN ON THE MAINTENANCE REGISTER.
016644  005701          MS2:    TST     R1
016646  005012                  CLR     (R2)

016650          MS3:                                    ;REPORT ERROR.  MAINTENANCE
016650  010437  001230          MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                        ;CAUSE ERROR.
016654  104127          1$:     ERROR   127
016656  012737  177777  030760  MOV     #-1,MANFL2
016664  000500                  BR      MSDONE

016666  022737  023404  177744  MSERR0: CMP     #23404,@#MEMERR ;DID THE ERROR REGISTER
016674  001042                  BNE     69$             ;SET PROPERLY?

016676  022626          64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
016700  005037  177572  65$:    CLR     @#MMR0
016704  005037  172516          CLR     @#MMR3
016710  012737  177777  177744  MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
016716  005737  177744          TST     @#MEMERR        ;REGISTER.
016722  001416                  BEQ     68$
```

```
 4813  016724                          66$:                                    ;ERROR REGISTER WON'T
 4814  016724  013737  177740  001230          MOV     @#LOADRS,$TMP2          ;CLEAR
 4815  016732  013737  177742  001232          MOV     @#HIADRS,$TMP3
 4816  016740  013737  177744  001234          MOV     @#MEMERR,$TMP4
 4817
 4818  016746  104130                  67$:    ERROR   130
 4819  016750  012737  177777  030740          MOV     #-1,MMRFLG              ;SIGNAL BAD REGISTER
 4820  016756  000443                          BR      MSDONE
 4821
 4822  016760  022737  177740  177740  68$:    CMP     #177740,@#LOADRS        ;SEE IF ADDRESS REGISTER
 4823  016766  001356                          BNE     66$                     ;UNLOCKED.
 4824  016770  022737  000003  177742          CMP     #3,@#HIADRS
 4825  016776  001352                          BNE     66$
 4826  017000  000432                          BR      MSDONE
 4827
 4828  017002                          69$:                                    ;REPORT ERROR REGISTER
 4829  017002  012637  001230                  MOV     (SP)+,$TMP2             ;NOT SET AS EXPECTED.
 4830  017006  005726                          TST     (SP)+                   ;RESET THE STACK.
 4831  017010  013737  177740  001232          MOV     @#LOADRS,$TMP3
 4832  017016  013737  177742  001234          MOV     @#HIADRS,$TMP4
 4833  017024  012737  010000  001236          MOV     #10000,$TMP5
 4834  017032  012737  023404  001240          MOV     #23404,$TMP6
 4835  017040  013737  177744  001242          MOV     @#MEMERR,$TMP7
 4836
 4837  017046  104131                  70$:    ERROR   131
 4838  017050  012737  177777  030760          MOV     #-1,MANFL2              ;SIGNAL BAD REGISTER
 4839  017056  012737  177777  030754          MOV     #-1,MMRFL2
 4840  017064  000705                          BR      65$
 4841  017066  104416          MSDONE: RSET
 4842
 4843                                  ;;**************************************************************
 4844                                  ;*TEST 36       CACHE MAINTENANCE AND ERROR REGISTERS TEST 22
 4845                                  ;*
 4846                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
 4847                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
 4848                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
 4849                                  ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
 4850                                  ;*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN ODD WORD IN THE
 4851                                  ;*PAIR, WHICH IS ALSO THE WANTED WORD.
 4852                                  ;*
 4853                                  ;;**************************************************************
 4854  017070  000004          TST36:  SCOPE
 4855  017072  012737  000040  001274          MOV     #40,$TIMES              ;;DO 40 ITERATIONS
 4856           000036          MT=$TN-1
 4857                                                                          ;SET THE SKAD REGISTER
 4858  017100  012737  017514  030524          MOV     #TST37,SKAD             ;IN CASE THE TEST ABORTS.
 4859
 4860  017106  113737  001102  001224          MOVB    $TSTNM,$TMP0
 4861
 4862  017114  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
 4863  017116  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
 4864  017120  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
 4865  017122  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
 4866  017124  104422                          MMSKIP
 4867
 4868  017126  012700  172340                  MOV     #KIPAR0,R0              ;SET UP MEMORY MANAGEMENT
```

```
                                                                    ;TO RELOCATE EVERYTHING
017132  012702  172300                  MOV     #KIPDR0,R2          ;THROUGH THE UNIBUS
017136  012703  000007                  MOV     #7,R3               ;MAP PASSIVELY TO MEMORY,
017142  005004                          CLR     R4                  ;BY PASSIVELY IS MEANT
017144  012705  170200                  MOV     #MAPL00,R5          ;THAT ADDRESS ARE
                                                                    ;RELOCATED TO THEMSELVES.
017150  012722  077406          64$:    MOV     #77406,(R2)+
017154  010401                          MOV     R4,R1
017156  072127  000006                  ASH     #6,R1
017162  010125                          MOV     R1,(R5)+
017164  005025                          CLR     (R5)+
017166  010410                          MOV     R4,(R0)
017170  062720  170000                  ADD     #170000,(R0)+
017174  062704  000200                  ADD     #200,R4
017200  077315                          SOB     R3,64$
017202  012710  177600                  MOV     #177600,(R0)
017206  012712  077406                  MOV     #77406,(R2)

017212  012737  000060  172516          MOV     #60,@#MMR3          ;TURN ON THE MAP AND 22-BIT
017220  012737  000001  177572          MOV     #1,@#MMR0           ;MODE ADDRESSING.
017226  012737  017312  000114          MOV     #MTERR0,@#CACHVEC           ;SET UP FOR THE ERROR.
017234  012737  000014  177746          MOV     #M0M1,@#CONTRL      ;FORCE MISSES TO BOTH GROUPS.
017242  012704  040000                  MOV     #40000,R4           ;PATTERN TO BE PUT IN MAINT.
017246  012702  177750                  MOV     #MAINT,R2           ;REG.
017252  000403                          BR      MT1

        017254                          LOC=.               ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        017254                          LOC=-4&LOC
        017260                          LOC=LOC+4
        017260                          .=LOC

017260  000240                          NOP
017262  000240                  MT1:    NOP                         ;NOP FOR SCOPING WITH AN OSCILLOSCOPE!!
017264  010412                          MOV     R4,(R2)             ;SET THE MAINT. REG.
017266  005701                          TST     R1                  ;THE REFERENCE TO THIS INSTRUCTION SHOULD CAUSE A PARITY
017270  005012                          CLR     (R2)                ;ABORT CAUSED BY DETECTION OF BAD PARITY ON
017272  000240                          NOP                         ;THE WANTED, ODD, WORD IN THIS PAIR.


017274                          MT2:                                ;REPORT ERROR.  MAINTENANCE
017274  010437  001230                  MOV     R4,$TMP2            ;FUNCTION FAILED TO
                                                                    ;CAUSE ERROR.
017300  104127                  1$:     ERROR   127
017302  012737  177777  030760  MOV     #-1,MANFL2
017310  000500                          BR      MTDONE

017312  022737  023410  177744  MTERR0: CMP     #23410,@#MEMERR     ;DID THE ERROR REGISTER
017320  001042                          BNE     69$                 ;SET PROPERLY?

017322  022626                  64$:    CMP     (SP)+,(SP)+         ;RESET THE STACK
017324  005037  177572          65$:    CLR     @#MMR0
017330  005037  172516                  CLR     @#MMR3
017334  012737  177777  177744          MOV     #-1,@#MEMERR        ;TRY TO CLEAR THE ERROR
017342  005737  177744                  TST     @#MEMERR            ;REGISTER.
017346  001416                          BEQ     68$
```

```
                                66$:                                    ;ERROR REGISTER WON'T
017350  013737  177740  001230        MOV     @#LOADRS,$TMP2            ;CLEAR
017356  013737  177742  001232        MOV     @#HIADRS,$TMP3
017364  013737  177744  001234        MOV     @#MEMERR,$TMP4

017372  104130                67$:    ERROR   130
017374  012737  177777  030740        MOV     #-1,MMRFLG                ;SIGNAL BAD REGISTER
017402  000443                        BR      MTDONE

017404  022737  177740  177740  68$:  CMP     #177740,@#LOADRS          ;SEE IF ADDRESS REGISTER
017412  001356                        BNE     66$                       ;UNLOCKED.
017414  022737  000003  177742        CMP     #3,@#HIADRS
017422  001352                        BNE     66$
017424  000432                        BR      MTDONE

017426                          69$:                                    ;REPORT ERROR REGISTER
017426  012637  001230                MOV     (SP)+,$TMP2               ;NOT SET AS EXPECTED.
017432  005726                        TST     (SP)+                     ;RESET THE STACK.
017434  013737  177740  001232        MOV     @#LOADRS,$TMP3
017442  013737  177742  001234        MOV     @#HIADRS,$TMP4
017450  012737  040000  001236        MOV     #40000,$TMP5
017456  012737  023410  001240        MOV     #23410,$TMP6
017464  013737  177744  001242        MOV     @#MEMERR,$TMP7

017472  104131                70$:    ERROR   131
017474  012737  177777  030760        MOV     #-1,MANFL2                ;SIGNAL BAD REGISTER
017502  012737  177777  030754        MOV     #-1,MMRFL2
017510  000705                        BR      65$
017512  104416                MTDONE: RSET

                                ;;**********************************************************************
                                ;*TEST 37        CACHE MAINTENANCE AND ERROR REGISTERS TEST 23
                                ;*
                                ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
                                ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
                                ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
                                ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY
                                ;*PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE
                                ;*LOW BYTE OF THAT ADDRESS .
                                ;*
                                ;**********************************************************************
000004                  TST37:  SCOPE
012737  000040  001274          MOV     #40,$TIMES          ;;DO 40 ITERATIONS
000037                  MU=$TN-1
                                                            ;SET THE SKAD REGISTER
012737  020134  030524          MOV     #TST40,SKAD         ;IN CASE THE TEST ABORTS.

113737  001102  001224          MOVB    $TSTNM,$TMP0

104430                          SKPBER                      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
104432                          SKPBCN                      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
104434                          SKPBMN                      ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
104436                          SKPBHM                      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
104422                          MMSKIP

012700  172340                  MOV     #KIPAR0,R0          ;SET UP MEMORY MANAGEMENT
```

```
                                                        ;TO RELOCATE EVERYTHING
017556  012702  172300                MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
017562  012703  000007                MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
017566  005004                        CLR     R4              ;BY PASSIVELY IS MEANT
017570  012705  170200                MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                        ;RELOCATED TO THEMSELVES.
017574  012722  077406        64$:    MOV     #77406,(R2)+
017600  010401                        MOV     R4,R1
017602  072127  000006                ASH     #6,R1
017606  010125                        MOV     R1,(R5)+
017610  005025                        CLR     (R5)+
017612  010410                        MOV     R4,(R0)
017614  062720  170000                ADD     #170000,(R0)+
017620  062704  000200                ADD     #200,R4
017624  077315                        SOB     R3,64$
017626  012710  177600                MOV     #177600,(R0)
017632  012712  077406                MOV     #77406,(R2)

017636  012737  000060  172516        MOV     #60,@#MMR3      ;TURN ON THE MAP AND
017644  012737  000001  177572        MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
017652  012737  017732  000114        MOV     #MUERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
017660  012737  000030  177746        MOV     #SOM1,@#CONTRL  ;SELECT GROUP ADDRESS
017666  012704  000400                MOV     #400,R4 ;PATTERN TO BE LOADED IN THE
017672  012702  177750                MOV     #MAINT,R2       ;MAINTENANCE REG.
017676  000403                        BR      MU1

                017700                  LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                017700                  LOC=-4$LOC
                017704                  LOC=LOC+4
                017704                  .=LOC

017704  000240                  MU1:    NOP
017706  000240                  MU2:    NOP
017710  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
017712  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
                                                        ;A PARITY ERROR IN GROUP
                                                        ;ADDRESS 0 MEMORY

017714                          MU2:                            ;REPORT ERROR.  MAINTENANCE
017714  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                        ;CAUSE ERROR.
017720  104127                  1$:     ERROR   127
017722  012737  177777  030760  MOV     #-1,MANFL2
017730  000500                          BR      MUDONE

017732  022737  002420  177744  MUERR0: CMP     #2420,@#MEMERR  ;DID THE ERROR REGISTER
017740  001042                          BNE     69$             ;SET PROPERLY?

017742  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
017744  005037  177572          65$:    CLR     @#MMR0
017750  005037  172516                  CLR     @#MMR3
017754  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
017762  005737  177744                  TST     @#MEMERR        ;REGISTER.
017766  001416                          BEQ     68$

017770                          66$:                            ;ERROR REGISTER WON'T
```

```
5037  017770  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
5038  017776  013737  177742  001232          MOV     @#HIADRS,$TMP3
5039  020004  013737  177744  001234          MOV     @#MEMERR,$TMP4
5040
5041  020012  104130                  67$:    ERROR   130
5042  020014  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
5043  020022  000443                          BR      MUDONE
5044
5045  020024  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
5046  020032  001356                          BNE     66$             ;UNLOCKED.
5047  020034  022737  000003  177742          CMP     #3,@#HIADRS
5048  020042  001352                          BNE     66$
5049  020044  000432                          BR      MUDONE
5050
5051  020046                          69$:                            ;REPORT ERROR REGISTER
5052  020046  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
5053  020052  005726                          TST     (SP)+           ;RESET THE STACK.
5054  020054  013737  177740  001232          MOV     @#LOADRS,$TMP3
5055  020062  013737  177742  001234          MOV     @#HIADRS,$TMP4
5056  020070  012737  000400  001236          MOV     #400,$TMP5
5057  020076  012737  002420  001240          MOV     #2420,$TMP6
5058  020104  013737  177744  001242          MOV     @#MEMERR,$TMP7
5059
5060  020112  104131                  70$:    ERROR   131
5061  020114  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
5062  020122  012737  177777  030754          MOV     #-1,MMRFL2
5063  020130  000705                          BR      65$
5064  020132  104416                  MUDONE: RSET
5065
5066                                  ;;*****************************************************************
5067                                  ;*TEST 40       CACHE MAINTENANCE AND ERROR REGISTERS TEST 24
5068                                  ;*
5069                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
5070                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
5071                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
5072                                  ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY
5073                                  ;*PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE
5074                                  ;*LOW BYTE OF THAT ADDRESS .
5075                                  ;*
5076                                  ;;*****************************************************************
5077  020134  000004          TST40:  SCOPE
5078  020136  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
5079           000040          MV=$TN-1
5080                                                                  ;SET THE SKAD REGISTER
5081  020144  012737  020554  030524          MOV     #TST41,SKAD     ;IN CASE THE TEST ABORTS.
5082
5083  020152  113737  001102  001224          MOVB    $TSTNM,$TMP0
5084
5085  020160  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
5086  020162  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
5087  020164  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
5088  020166  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
5089  020170  104422                          MMSKIP
5090
5091  020172  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
5092                                                                  ;TO RELOCATE EVERYTHING
```

```
5093  020176  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
5094  020202  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
5095  020206  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
5096  020210  012705  170200                  MOV     #MAPLOD,R5      ;THAT ADDRESS ARE
5097                                                                  ;RELOCATED TO THEMSELVES.
5098  020214  012722  077406          64$:    MOV     #77406,(R2)+
5099  020220  010401                          MOV     R4,R1
5100  020222  072127  000006                  ASH     #6,R1
5101  020226  010125                          MOV     R1,(R5)+
5102  020230  005025                          CLR     (R5)+
5103  020232  010410                          MOV     R4,(R0)
5104  020234  062720  170000                  ADD     #170000,(R0)+
5105  020240  062704  000200                  ADD     #200,R4
5106  020244  077315                          SOB     R3,64$
5107  020246  012710  177600                  MOV     #177600,(R0)
5108  020252  012712  077406                  MOV     #77406,(R2)
5109
5110  020256  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
5111  020264  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
5112  020272  012737  020352  000114          MOV     #MVERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
5113  020300  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;SELECT GROUP ADDRESS
5114  020306  012704  002000                  MOV     #2000,R4        ;PATTERN TO BE LOADED IN THE
5115  020312  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REG.
5116  020316  000403                          BR      MV1
5117
5118          020320                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5119          020320                          LOC=-4&LOC
5120          020324                          LOC=LOC+4
5121          020324                          .=LOC
5122
5123  020324  000240                          NOP
5124  020326  000240                  MV1:    NOP
5125  020330  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
5126  020332  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
5127                                                                  ;A PARITY ERROR IN GROUP
5128                                                                  ;ADDRESS 1 MEMORY
5129
5130  020334                          MV2:                            ;REPORT ERROR.  MAINTENANCE
5131  020334  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
5132                                                                  ;CAUSE ERROR.
5133  020340  104127                  1$:     ERROR   127
5134  020342  012737  177777  030760  MOV     #-1,MANFL2
5135  020350  000500                          BR      MVDONE
5136
5137  020352  022737  002440  177744  MVERR0: CMP     #2440,@#MEMERR  ;DID THE ERROR REGISTER
5138  020360  001042                          BNE     69$             ;SET PROPERLY?
5139
5140  020362  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
5141  020364  005037  177572          65$:    CLR     @#MMR0
5142  020370  005037  172516                  CLR     @#MMR3
5143  020374  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
5144  020402  005737  177744                  TST     @#MEMERR        ;REGISTER.
5145  020406  001416                          BEQ     68$
5146
5147  020410                          66$:                            ;ERROR REGISTER WON'T
5148  020410  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
```

```
5149  020416  013737  177742  001232          MOV     @#HIADRS,$TMP3
5150  020424  013737  177744  001234          MOV     @#MEMERR,$TMP4
5151
5152  020432  104130                  67$:    ERROR   130
5153  020434  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
5154  020442  000443                          BR      MVDONE
5155
5156  020444  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
5157  020452  001356                          BNE     66$             ;UNLOCKED.
5158  020454  022737  000003  177742          CMP     #3,@#HIADRS
5159  020462  001352                          BNE     66$
5160  020464  000432                          BR      MVDONE
5161
5162  020466                          69$:                            ;REPORT ERROR REGISTER
5163  020466  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
5164  020472  005726                          TST     (SP)+           ;RESET THE STACK.
5165  020474  013737  177740  001232          MOV     @#LOADRS,$TMP3
5166  020502  013737  177742  001234          MOV     @#HIADRS,$TMP4
5167  020510  012737  002000  001236          MOV     #2000,$TMP5
5168  020516  012737  002440  001240          MOV     #2440,$TMP6
5169  020524  013737  177744  001242          MOV     @#MEMERR,$TMP7
5170
5171  020532  104131                  70$:    ERROR   131
5172  020534  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
5173  020542  012737  177777  030754          MOV     #-1,MMRFL2
5174  020550  000705                          BR      65$
5175  020552  104416              MVDONE:     RSET
5176
5177                                  ;;**************************************************************
5178                                  ;*TEST 41        CACHE MAINTENANCE AND ERROR REGISTERS TEST 25
5179                                  ;*
5180                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
5181                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
5182                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
5183                                  ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY
5184                                  ;*PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE
5185                                  ;*LOW BYTE OF THAT DATA .
5186                                  ;*
5187                                  ;**************************************************************
5188  020554  000004              TST41:  SCOPE
5189  020556  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
5190          000041              MW=$TN-1
5191                                                                  ;SET THE SKAD REGISTER
5192  020564  012737  021174  030524          MOV     #TST42,SKAD     ;IN CASE THE TEST ABORTS.
5193
5194  020572  113737  001102  001224          MOVB    $TSTNM,$TMP0
5195
5196  020600  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
5197  020602  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
5198  020604  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
5199  020606  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
5200  020610  104422                          MMSKIP
5201
5202  020612  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
5203                                                                  ;TO RELOCATE EVERYTHING
5204  020616  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
```

```
5205  020622  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
5206  020626  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
5207  020630  012705  170200                  MOV     #MAPLOC,R5      ;THAT ADDRESS ARE
5208                                                                  ;RELOCATED TO THEMSELVES.
5209  020634  012722  077406          64$:    MOV     #77406,(R2)+
5210  020640  010401                          MOV     R4,R1
5211  020642  072127  000006                  ASH     #6,R1
5212  020646  010125                          MOV     R1,(R5)+
5213  020650  005025                          CLR     (R5)+
5214  020652  010410                          MOV     R4,(R0)
5215  020654  062720  170000                  ADD     #170000,(R0)+
5216  020660  062704  000200                  ADD     #200,R4
5217  020664  077315                          SOB     R3,64$
5218  020666  012710  177600                  MOV     #177600,(R0)
5219  020672  012712  077406                  MOV     #77406,(R2)
5220
5221  020676  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
5222  020704  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
5223  020712  012737  020772  000114          MOV     #MWERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
5224  020720  012737  000030  177746          MOV     #SOM1,@#CNTRL   ;SELECT GROUP DATA
5225  020726  012704  000020                  MOV     #20,R4  ;PATTERN TO BE LOADED IN THE
5226  020732  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REG.
5227  020736  000403                          BR      MW1
5228
5229          020740                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5230          020740                          LOC=-4&LOC
5231          020744                          LOC=LOC+4
5232          020744                          .=LOC
5233
5234  020744  000240                          NOP
5235  020746  000240          MW1:            NOP
5236  020750  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
5237  020752  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
5238                                                                  ;A PARITY ERROR IN GROUP
5239                                                                  ;DATA 0 MEMORY
5240
5241  020754          MW2:                                            ;REPORT ERROR.  MAINTENANCE
5242  020754  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
5243                                                                  ;CAUSE ERROR.
5244  020760  104127          1$:     ERROR   127
5245  020762  012737  177777  030760  MOV     #-1,MANFL2
5246  020770  000500                          BR      MWDONE
5247
5248  020772  022737  002500  177744  MWERR0: CMP     #2500,@#MEMERR  ;DID THE ERROR REGISTER
5249  021000  001042                          BNE     69$             ;SET PROPERLY?
5250
5251  021002  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
5252  021004  005037  177572          65$:    CLR     @#MMR0
5253  021010  005037  172516                  CLR     @#MMR3
5254  021014  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
5255  021022  005737  177744                  TST     @#MEMERR        ;REGISTER.
5256  021026  001416                          BEQ     68$
5257
5258  021030                  66$:                                    ;ERROR REGISTER WON'T
5259  021030  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
5260  021036  013737  177742  001232          MOV     @#HIADRS,$TMP3
```

```
5261  021044  013737  177744  001234          MOV     @#MEMERR,$TMP4
5262
5263  021052  104130                  67$:    ERROR   130
5264  021054  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
5265  021062  000443                          BR      MWDONE
5266
5267  021064  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
5268  021072  001356                          BNE     66$             ;UNLOCKED.
5269  021074  022737  000003  177742          CMP     #3,@#HIADRS
5270  021102  001352                          BNE     66$
5271  021104  000432                          BR      MWDONE
5272
5273  021106                      69$:                                ;REPORT ERROR REGISTER
5274  021106  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
5275  021112  005726                          TST     (SP)+           ;RESET THE STACK.
5276  021114  013737  177740  001232          MOV     @#LOADRS,$TMP3
5277  021122  013737  177742  001234          MOV     @#HIADRS,$TMP4
5278  021130  012737  000020  001236          MOV     #20,$TMP5
5279  021136  012737  002500  001240          MOV     #2500,$TMP6
5280  021144  013737  177744  001242          MOV     @#MEMERR,$TMP7
5281
5282  021152  104131                  70$:    ERROR   131
5283  021154  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
5284  021162  012737  177777  030754          MOV     #-1,MMRFL2
5285  021170  000705                          BR      65$
5286  021172  104416              MWDONE:     RSET
5287
5288                              ;;**************************************************************
5289                              ;*TEST 42       CACHE MAINTENANCE AND ERROR REGISTERS TEST 26
5290                              ;*
5291                              ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
5292                              ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
5293                              ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
5294                              ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY
5295                              ;*PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE
5296                              ;*LOW BYTE OF THAT DATA .
5297                              ;*
5298                              ;;**************************************************************
5299  021174  000004              TST42:  SCOPE
5300  021176  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
5301          000042              MX=$TN-1
5302                                                                  ;SET THE SKAD REGISTER
5303  021204  012737  021614  030524          MOV     #TST43,SKAD     ;IN CASE THE TEST ABORTS.
5304
5305  021212  113737  001102  001224          MOVB    $TSTNM,$TMP0
5306
5307  021220  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
5308  021222  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
5309  021224  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
5310  021226  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
5311  021230  104422                          MMSKIP
5312
5313  021232  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
5314                                                                  ;TO RELOCATE EVERYTHING
5315  021236  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
5316  021242  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
```

```
5317  021246  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
5318  021250  012705  170200                  MOV     #MAPLOO,R5      ;THAT ADDRESS ARE
5319                                                                  ;RELOCATED TO THEMSELVES.
5320  021254  012722  077406          64$:    MOV     #77406,(R2)+
5321  021260  010401                          MOV     R4,R1
5322  021262  072127  000006                  ASH     #6,R1
5323  021266  010125                          MOV     R1,(R5)+
5324  021270  005025                          CLR     (R5)+
5325  021272  010410                          MOV     R4,(R0)
5326  021274  062720  170000                  ADD     #170000,(R0)+
5327  021300  062704  000200                  ADD     #200,R4
5328  021304  077315                          SOB     R3,64$
5329  021306  012710  177600                  MOV     #177600,(R0)
5330  021312  012712  077406                  MOV     #77406,(R2)
5331
5332  021316  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
5333  021324  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
5334  021332  012737  021412  000114          MOV     #MXERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
5335  021340  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;SELECT GROUP DATA
5336  021346  012704  000100                  MOV     #100,R4 ;PATTERN TO BE LOADED IN THE
5337  021352  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REG.
5338  021356  000403                          BR      MX1
5339
5340          021360                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5341          021360                          LOC=-4&LOC
5342          021364                          LOC=LOC+4
5343          021364                          .=LOC
5344
5345  021364  000240                          NOP
5346  021366  000240          MX1:            NOP
5347  021370  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
5348  021372  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
5349                                                                  ;A PARITY ERROR IN GROUP
5350                                                                  ;DATA 1 MEMORY
5351
5352  021374                  MX2:                                    ;REPORT ERROR.  MAINTENANCE
5353  021374  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
5354                                                                  ;CAUSE ERROR.
5355  021400  104127          1$:     ERROR   127
5356  021402  012737  177777  030760  MOV     #-1,MANFL2
5357  021410  000500                          BR      MXDONE
5358
5359  021412  022737  002600  177744  MXERR0: CMP     #2600,@#MEMERR  ;DID THE ERROR REGISTER
5360  021420  001042                          BNE     69$             ;SET PROPERLY?
5361
5362  021422  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
5363  021424  005037  177572          65$:    CLR     @#MMR0
5364  021430  005037  172516                  CLR     @#MMR3
5365  021434  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
5366  021442  005737  177744                  TST     @#MEMERR        ;REGISTER.
5367  021446  001416                          BEQ     68$
5368
5369  021450                  66$:                                    ;ERROR REGISTER WON'T
5370  021450  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
5371  021456  013737  177742  001232          MOV     @#HIADRS,$TMP3
5372  021464  013737  177744  001234          MOV     @#MEMERR,$TMP4
```

```
5373
5374  021472  104130                  67$:    ERROR   130
5375  021474  012737  177777  030740          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
5376  021502  000443                          BR      MXDONE
5377
5378  021504  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
5379  021512  001356                          BNE     66$             ;UNLOCKED.
5380  021514  022737  000003  177742          CMP     #3,@#HIADRS
5381  021522  001352                          BNE     66$
5382  021524  000432                          BR      MXDONE
5383
5384  021526                          69$:                            ;REPORT ERROR REGISTER
5385  021526  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
5386  021532  005726                          TST     (SP)+           ;RESET THE STACK.
5387  021534  013737  177740  001232          MOV     @#LOADRS,$TMP3
5388  021542  013737  177742  001234          MOV     @#HIADRS,$TMP4
5389  021550  012737  000100  001236          MOV     #100,$TMP5
5390  021556  012737  002600  001240          MOV     #2600,$TMP6
5391  021564  013737  177744  001242          MOV     @#MEMERR,$TMP7
5392
5393  021572  104131                  70$:    ERROR   131
5394  021574  012737  177777  030760          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
5395  021602  012737  177777  030754          MOV     #-1,MMRFL2
5396  021610  000705                          BR      65$
5397  021612  104416                  MXDONE: RSET
5398
5399                                  ;;**********************************************************************
5400                                  ;*TEST 43       CACHE ERROR REGISTER UNIBUS TIME OUT TEST
5401                                  ;*
5402                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO COMPREHEND A
5403                                  ;*CPU TO UNIBUS THROUGH THE MAP TO THE CACHE REFERENCE WHICH
5404                                  ;*TIMES OUT IN MAIN MEMORY. MANY SUCH NON-EXISTENT MEMORY LOCATIONS
5405                                  ;*ARE CONVIENTLY GUARENTEED TO EXIST! ALL THE ADDRESSES
5406                                  ;*FROM 17000000 THROUGH 17777776 ARE ADDRESSES
5407                                  ;*WHICH CAN NOT EXIST. HERE ONLY ONE OF THESE ADDRESSES, 17777776,
5408                                  ;*WILL BE USED TO CAUSE A TIME OUT ON THE UNIBUS AN THE CONSEQUENT
5409                                  ;*ABORT TO VECTOR ERRVEC.
5410                                  ;*
5411                                  ;;**********************************************************************
5412  021614  000004                  TST43:  SCOPE
5413  021616  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
5414          000043                  MQ=$TN-1
5415                                                                  ;SET THE SKAD REGISTER
5416  021624  012737  022244  030524          MOV     #TST44,SKAD     ;IN CASE THE TEST ABORTS.
5417
5418  021632  113737  001102  001224          MOVB    $TSTNM,$TMP0
5419  021640  012737  030400  000114          MOV     #SPUR,@#CACHVEC         ;EXPECT NO PARITY ERRORS.
5420
5421  021646  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
5422  021650  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
5423  021652  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
5424  021654  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
5425  021656  104422                          MMSKIP
5426
5427  021660  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
5428  021664  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
```

```
5429  021670  012702  172300              MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
5430  021674  012703  000010              MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
5431  021700  010122              64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
5432  021702  077302                      SOB     R3,64$          ;ADDRESSES.
5433  021704  005020                      CLR     (R0)+
5434  021706  012720  000200              MOV     #200,(R0)+
5435  021712  012720  000400              MOV     #400,(R0)+
5436  021716  012720  000600              MOV     #600,(R0)+
5437  021722  012720  001000              MOV     #1000,(R0)+
5438  021726  012720  001200              MOV     #1200,(R0)+
5439  021732  012720  001400              MOV     #1400,(R0)+
5440  021736  012710  177600              MOV     #177600,(R0)
5441
5442  021742  012737  000060  172516      MOV     #60,@#MMR3              ;TURN ON THE MAPPING BOX
5443  021750  012737  000001  177572      MOV     #1,@#MMR0               ;AND 22 BIT MODE ADDRESSING.
5444  021756  012737  170000  172354      MOV     #170000,@#KIPAR6        ;MAKE KIPAR6 RELOCATE
5445                                                                      ;TO THE UNIBUS.
5446  021764  012737  022036  000004      MOV     #MQERR,@#ERRVEC         ;SET UP THE TIME OUT VECTOR.
5447
5448  021772  012737  177776  170200      MOV     #-2,@#MAPL00            ;SET THE MAP REGISTER 0
5449  022000  012737  000077  170202      MOV     #77,@#MAPH00
5450  022006  012700  140000              MOV     #140000,R0              ;THIS IS THE VIRTUAL ADDRESS OF THE
5451                                                                      ;TEST ADDRESS. IT WILL RELOCATE
5452                                                                      ;THROUGH KIPAR6 TO THE UNIBUS AS
5453                                                                      ;A 000000. FROM THE UNIBUS
5454                                                                      ;IT WILL BE RELOCATED THROUGH
5455                                                                      ;MAP REGISTER 0 TO THE CACHE WHERE
5456                                                                      ;IT WILL TRY TO REFERENCE
5457                                                                      ;17777776, AND HOPEFULLY TIME OUT.
5458  022012  000240              NOP                                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
5459  022014  005710              TST     (R0)                            ;MAKE THE REFERENCE!
5460
5461  022016              MQ1:                                            ;NO TIME OUT OCCURRED, REPORT
5462  022016  012737  177776  001230      MOV     #-2,$TMP2               ;THE ERROR.
5463  022024  012737  000077  001232      MOV     #77,$TMP3
5464  022032  104132              1$:     ERROR   132
5465  022034  000502                      BR      MQDONE
5466
5467  022036  032737  000020  177766 MQERR: BIT    #20,@#CPUERR            ;SEE IF A TIME OUT HAS CAUSED
5468  022044  001002                      BNE     MQ2                     ;AN ABORT TO THIS ROUTINE.
5469  022046  000137  030352              JMP     CPSPUR                  ;IF NOT GO TO THE SPURIOUS
5470                                                                      ;UNEXPECTED, CPU ERROR HANDLER.
5471  022052  022737  000000  177744 MQ2:  CMP     #0,@#MEMERR             ;OTHERWISE SEE IF THE ERROR
5472  022060  001427                      BEQ     MQ3                     ;REGISTER GOT SET CORRECTLY.
5473
5474                                                              ;IF IT IS NOT SET CORRECTLY REPORT ERROR.
5475  022062  012637  001230              MOV     (SP)+,$TMP2
5476  022066  005726                      TST     (SP)+
5477  022070  013737  177740  001232      MOV     @#LOADRS,$TMP3
5478  022076  013737  177742  001234      MOV     @#HIADRS,$TMP4
5479  022104  012737  177776  001236      MOV     #-2,$TMP5
5480  022112  012737  000077  001240      MOV     #77,$TMP6
5481  022120  013737  177744  001242      MOV     @#MEMERR,$TMP7
5482  022126  104133              1$:     ERROR   133
5483  022130  012737  177777  030754      MOV     #-1,MMRFL2
5484  022136  000401                      BR      MQ4
```

```
5485
5486  022140  022626                    MQ3:    CMP     (SP)+,(SP)+             ;RESET THE STACK
5487
5488  022142  005037  177572            MQ4:    CLR     @#MMR0
5489  022146  005037  172516                    CLR     @#MMR3
5490  022152  012737  177777  177744            MOV     #-1,@#MEMERR            ;TRY TO CLEAR THE ERROR REGISTER.
5491  022160  005737  177744                    TST     @#MEMERR
5492  022164  001416                            BEQ     MQ6
5493
5494  022166                    MQ5:                                    ;REPORT THE FAILURE OF THE ERROR
5495  022166  013737  177740  001230            MOV     @#LOADRS,$TMP2          ;REGISTER TO CLEAR!
5496  022174  013737  177742  001232            MOV     @#HIADRS,$TMP3
5497  022202  013737  177744  001234            MOV     @#MEMERR,$TMP4
5498  022210  104130                    1$:     ERROR   130
5499  022212  012737  177777  030740            MOV     #-1,MMRFLG
5500  022220  000410                            BR      MQDONE
5501
5502  022222  022737  177740  177740  MQ6:      CMP     #177740,@#LOADRS        ;SEE IF THE ADDRESS REGISTER
5503  022230  001356                            BNE     MQ5                     ;GOT RESET.
5504  022232  022737  000003  177742            CMP     #3,@#HIADRS
5505  022240  001352                            BNE     MQ5
5506
5507  022242  104416            MQDONE: RSET
5508
5509                            ;;**************************************************************
5510                            ;*TEST 44        CACHE CONTROL REGISTER DISABLE TRAPS TEST 1
5511                            ;*
5512                            ;*THIS IS A TEST OF THE CONTROL REGISTER'S ABILITY TO DISABLE A TRAP
5513                            ;*OCCURRING AS THE RESULT OF A MAIN MEMORY DATA PARITY ERROR IN THE
5514                            ;*UNWANTED WORD OF THE REFERENCED PAIR. THE MAINTENANCE REGISTER IS
5515                            ;*USED TO FORCE AN ERROR ON THE LOW BYTE OF THE ODD WORD WHEN REFERENCING
5516                            ;*THE EVEN WORD OF THAT PAIR.
5517                            ;*
5518                            ;;**************************************************************
5519  022244  000004            TST44:  SCOPE
5520  022246  012737  000040  001274            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
5521          000044            KV=$TN-1
5522                                                    ;SET THE SKAD REGISTER
5523  022254  012737  022420  030524            MOV     #TST45,SKAD     ;IN CASE THE TEST ABORTS.
5524
5525  022262  113737  001102  001224            MOVB    $TSTNM,$TMP0
5526
5527  022270  104430                            SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
5528  022272  104432                            SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
5529  022274  104434                            SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
5530  022276  104436                            SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
5531  022300  012737  000014  177746            MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
5532  022306  052737  000001  177746            BIS     #BIT0,@#CONTRL          ;DISABLE 'WARNING' TRAPS.
5533  022314  012737  022356  000114            MOV     #KVERR,@#CACHVEC                ;SET UP FOR THE ERROR ABOUT TO BE FORCED
5534  022322  012704  040000                    MOV     #40000,R4               ;PATTERN FOR THE MAINTENANCE
5535  022326  012702  177750                    MOV     #MAINT,R2               ;REGISTER.
5536  022332  000402                            BR      KV1
5537
5538          022334                    LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5539          022334                    LOC=-4&LOC
5540          022340                    LOC=LOC+4
```

```
                022340                          .=LOC
                022340  000240          KV1:    NOP
                022342  010412                  MOV     R4,(R2)                 ;SET THE MAINTENANCE REGISTER
                022344  000240                  NOP                             ;WHEN THIS NOP IS FETCHED AN ERROR
                022346  005701          KV2:    TST     R1                      ;WILL BE RECOGNIZED BECAUSE OF THE
                                                                                ;CONTENTS OF THE LOCATION KV2!
                                                                                ;THIS PARITY ERROR WOULD
                                                                                ;NORMALLY RELUT IN A TRAP BUT
                                                                                ;BECAUSE TRAPS HAVE BEEN DISABLED
                                                                                ;NONE SHOULD OCCUR!!!
                022350  005012                  CLR     (R2)
                022352  000240                  NOP
                022354  000420                  BR      KVDONE                  ;GOOD, NO TRAP OCCURRED!

                022356                  KVERR:                          ;COME HERE IF A TRAP OCCURS
                022356  012637  001230          MOV     (SP)+,$TMP2             ;AND REPORT THE ERROR.
                022362  005726                  TST     (SP)+
                022364  013737  177746  001232  MOV     @#CONTRL,$TMP3
                022372  013737  177740  001234  MOV     @#LOADRS,$TMP4
                022400  013737  177742  001236  MOV     @#HIADRS,$TMP5
                022406  013737  177744  001240  MOV     @#MEMERR,$TMP6
                022414  104134          1$:     ERROR   134

                022416  104416          KVDONE: RSET


                                        ;;*********************************************************************
                                        ;*TEST 45       CACHE CONTROL REGISTER DISABLE TRAPS TEST 2
                                        ;*
                                        ;*THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION.
                                        ;*IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE ADDRESS
                                        ;*MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO
                                        ;*FORCE THE ERROR ON THE LOW BYTE OF THE ADDRESS, IN THE ADDRESS MEMORY
                                        ;*OF GROUP 0.
                                        ;*
                                        ;;*********************************************************************
                022420  000004  TST45:  SCOPE
                022422  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                        000045  KX=$TN-1
                                                                                ;SET THE SKAD REGISTER
                022430  012737  022620  030524          MOV     #TST46,SKAD     ;IN CASE THE TEST ABORTS.

                022436  113737  001102  001224          MOVB    $TSTNM,$TMP0

                022444  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
                022446  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
                022450  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
                022452  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
                022454  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;USE GROUP ZERO
                022462  012700  022550                  MOV     #KX2,R0         ;MAKE KX2 A HIT IN GROUP
                022466  005710                          TST     (R0)            ;ZERO.
                022470  005710                          TST     (R0)

                                                                                ;SEE IF REFERENCE ADDRESS
```

```
5597  022472  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
5598  022500  001007                          BNE     KX1
5599                                                                  ;IF NOT ERROR!
5600  022502  010037  001230                  MOV     R0,$TMP2
5601  022506  012737  000000  001226          MOV     #0,$TMP1
5602  022514  104001                          ERROR   1
5603
5604  022516  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
5605
5606  022520  052737  000001  177746  KX1:    BIS     #BIT0,@#CONTRL  ;DISABLE 'WARNING' TRAPS.
5607  022526  012737  022556  000114          MOV     #KXERR,@#CACHVEC        ;SET UP FOR ERROR WHICH
5608                                                                  ;SHOULD NOT TRAP!
5609  022534  012704  000400                  MOV     #400,R4         ;PATTERN FOR MAINT REG.
5610  022540  012702  177750                  MOV     #MAINT,R2
5611  022544  000240                          NOP
5612  022546  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
5613  022550  005012          KX2:            CLR     (R2)            ;THE FETCH OF THIS
5614  022552  000240                          NOP                     ;INSTRUCTION SHOULD CAUSE
5615  022554  000420                          BR      KXDONE          ;A CACHE  MEMORY
5616                                                                  ;PARITY ERROR WHICH
5617                                                                  ;NORMALLY SHOULD TRAP
5618                                                                  ;BUT HERE NO TRAP SHOULD
5619                                                                  ;OCCUR FOR TRAPS HAVE BEEN DISABLED.
5620
5621  022556          KXERR:                                          ;A TRAP HAS ERRONEOUSLY
5622  022556  012637  001230                  MOV     (SP)+,$TMP2     ;TAKEN PLACE, REPORT
5623  022562  005726                          TST     (SP)+           ;UNABLE TO DISABLE TRAPS.
5624  022564  013737  177746  001232          MOV     @#CONTRL,$TMP3
5625  022572  013737  177740  001234          MOV     @#LOADRS,$TMP4
5626  022600  013737  177742  001236          MOV     @#HIADRS,$TMP5
5627  022606  013737  177744  001240          MOV     @#MEMERR,$TMP6
5628
5629  022614  104134          1$:             ERROR   134
5630
5631  022616  104416          KXDONE: RSET
5632
5633
5634                                          ;;*****************************************************************
5635                                          ;*TEST 46       CACHE CONTROL REGISTER DISABLE TRAPS TEST 3
5636                                          ;*
5637                                          ;*THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION.
5638                                          ;*IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE
5639                                          ;*MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO
5640                                          ;*FORCE THE ERROR ON THE LOW BYTE OF THE , IN THE  MEMORY
5641                                          ;*OF GROUP 0.
5642                                          ;*
5643                                          ;;*****************************************************************
5644  022620  000004          TST46:  SCOPE
5645  022622  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
5646          000046          KZ=$TN-1
5647                                                                  ;SET THE SKAD REGISTER
5648  022630  012737  023020  030524          MOV     #TST47,SKAD     ;IN CASE THE TEST ABORTS.
5649
5650  022636  113737  001102  001224          MOVB    $TSTNM,$TMP0
5651
5652  022644  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
```

```
                                        SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
                                        SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
                                        SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
022646  104432
022650  104434
022652  104436
022654  012737  000030  177746         MOV     #SOM1,@#CONTRL  ;USE GROUP ZERO
022662  012700  022750                 MOV     #KZ2,R0         ;MAKE KZ2 A HIT IN GROUP
022666  005710                         TST     (R0)            ;ZERO.
022670  005710                         TST     (R0)
                                                                ;SEE IF REFERENCE ADDRESS
022672  032737  000010  177752         BIT     #10,@#HITMIS    ;IS A HIT.
022700  001007                         BNE     KZ1
                                                                ;IF NOT ERROR!
022702  010037  001230                 MOV     R0,$TMP2
022706  012737  000000  001226         MOV     #0,$TMP1
022714  104001                         ERROR   1
022716  104420                         SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
022720  052737  000001  177746  KZ1:   BIS     #BIT0,@#CONTRL  ;DISABLE 'WARNING' TRAPS.
022726  012737  022756  000114         MOV     #KZERR,@#CACHVEC        ;SET UP FOR ERROR WHICH
                                                                ;SHOULD NOT TRAP!
022734  012704  000020                 MOV     #20,R4          ;PATTERN FOR MAINT REG.
022740  012702  177750                 MOV     #MAINT,R2
022744  000240                         NOP
022746  010412                         MOV     R4,(R2)         ;SET THE MAINT. REG.
022750  005012          KZ2:           CLR     (R2)            ;THE FETCH OF THIS
022752  000240                         NOP                     ;INSTRUCTION SHOULD CAUSE
022754  000420                         BR      KZDONE          ;A CACHE  MEMORY
                                                                ;PARITY ERROR WHICH
                                                                ;NORMALLY SHOULD TRAP
                                                                ;BUT HERE NO TRAP SHOULD
                                                                ;OCCUR FOR TRAPS HAVE BEEN DISABLED.

022756          KZERR:                                          ;A TRAP HAS ERRONEOUSLY
022756  012637  001230                         MOV     (SP)+,$TMP2     ;TAKEN PLACE, REPORT
022762  005726                                 TST     (SP)+           ;UNABLE TO DISABLE TRAPS.
022764  013737  177746  001232                 MOV     @#CONTRL,$TMP3
022772  013737  177740  001234                 MOV     @#LOADRS,$TMP4
023000  013737  177742  001236                 MOV     @#HIADRS,$TMP5
023006  013737  177744  001240                 MOV     @#MEMERR,$TMP6

023014  104134          1$:            ERROR   134

023016  104416          KZDONE:        RSET


;;*****************************************************************
;*TEST 47       CACHE ERROR REGISTER LOCK UP TEST 1
;*
;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
```

```
                                        ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
                                        ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
                                        ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
                                        ;*TO THE CACHE DIRECTLY.
                                        ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
                                        ;*TO THE CACHE DIRECTLY.
                                        ;*
023020  000004                          ;;**************************************************************
                                TST47:  SCOPE
023022  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000047                  NA=$TN-1
                                                                ;SET THE SKAD REGISTER
023030  012737  023404  030524          MOV     #TST50,SKAD     ;IN CASE THE TEST ABORTS.

023036  113737  001102  001224          MOVB    $TSTNM,$TMP0

023044  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
023046  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
023050  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
023052  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
023054  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.

023062  012737  023136  000114          MOV     #NA3,@#CACHVEC          ;SET UP FOR THE ERROR.
023070  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
023074  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
023100  000401                          BR      NA1

        023102                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        023100                  LOC=-4&LOC
        023104                  LOC=LOC+4
        023104                  .=LOC

023104  000240                  NA1:    NOP
023106  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
023110  005701                  NA2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
023112  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
023114  000240                          NOP
                                                                ;IF NONE OCCURS REPORT
023116  012737  010000  001230          MOV     #10000,$TMP2    ;ERROR!
023124  104127                  1$:     ERROR   127
023126  012737  177777  030760          MOV     #-1,MANFL2
023134  000522                  BR      NADONE

023136                          NA3:

023136  012737  023212  000114          MOV     #NA6,@#CACHVEC          ;SET UP FOR THE ERROR.
023144  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
023150  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
023154  000401                          BR      NA4

        023156                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        023154                  LOC=-4&LOC
        023160                  LOC=LOC+4
        023160                  .=LOC
```

```
                                NA4:    NOP
                                        MOV     R4,(R2)         ;SET THE MAINT. REG.
                                NA5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
                                        CLR     (R2)            ;SHOULD CAUSE AN ABORT!
                                        NOP
                                                                ;IF NONE OCCURS REPORT
                                        MOV     #10000,$TMP2    ;ERROR!
                                1$:     ERROR   127
                                        MOV     #-1,MANFL2
                                BR      NADONE

                                NA6:

                                        ADD     #10,SP          ;RESET THE STACK.
                                        CMP     #144404,@#MEMERR        ;SEE IF THE ERROR REGISTER
                                        BNE     NA7             ;IS SET CORRECTLY.
                                        CMP     #NA2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
                                        BEQ     NA8             ;IS SET CORRECTLY.

                                NA7:                            ;NOT SET CORRECTLY!
                                        MOV     #144404,$TMP2   ;REPORT FAILURE.
                                        MOV     @#MEMERR,$TMP3
                                        MOV     #NA2,$TMP4
                                        CLR     $TMP5
                                        MOV     @#LOADRS,$TMP6
                                        MOV     @#HIADRS,$TMP7

                                1$:     ERROR   135

                                NA8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
                                        CLR     @#MMR3
                                        MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
                                        TST     @#MEMERR        ;ERROR REG.
                                        BEQ     NA10

                                NA9:                            ;WON'T CLEAR!
                                        MOV     @#LOADRS,$TMP2
                                        MOV     @#HIADRS,$TMP3
                                        MOV     @#MEMERR,$TMP4

                                1$:     ERROR   130
                                        MOV     #-1,MMRFLG
                                        BR      NADONE

                                NA10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
                                        BNE     NA9             ;HAS RESET
                                        CMP     #3,@#HIADRS
                                        BNE     NA9

                                NADONE: RSET


                                ;;*******************************************************************
                                ;*TEST 50        CACHE ERROR REGISTER LOCK UP TEST 2
```

```
                                        ;*
                                        ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
                                        ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
                                        ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
                                        ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
                                        ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
                                        ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
                                        ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
                                        ;*TO THE CACHE DIRECTLY.
                                        ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
                                        ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
                                        ;*
                                        ;;*****************************************************************
023404  000004                  TST50:  SCOPE
023406  012737  000040  001274          MOV     #40.,$TIMES     ;;DO 40 ITERATIONS
        000050          NB=$TN-1
                                                                ;SET THE SKAD REGISTER
023414  012737  024074  030524          MOV     #TST51,SKAD     ;IN CASE THE TEST ABORTS.

023422  113737  001102  001224          MOVB    $TSTNM,$TMPO

023430  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
023432  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
023434  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
023436  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
023440  104422                          MMSKIP

023442  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                ;TO RELOCATE EVERYTHING
023446  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
023452  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
023456  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
023460  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                                ;RELOCATED TO THEMSELVES.
023464  012722  077406          64$:    MOV     #77406,(R2)+
023470  010401                          MOV     R4,R1
023472  072127  000006                  ASH     #6,R1
023476  010125                          MOV     R1,(R5)+
023500  005025                          CLR     (R5)+
023502  010410                          MOV     R4,(R0)
023504  062720  170000                  ADD     #170000,(R0)+
023510  062704  000200                  ADD     #200,R4
023514  077315                          SOB     R3,64$
023516  012710  177600                  MOV     #177600,(R0)
023522  012712  077406                  MOV     #77406,(R2)

023526  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.


023534  012737  023612  000114          MOV     #NB3,@#CACHVEC          ;SET UP FOR THE ERROR.
023542  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
023546  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
023552  000402                          BR      NB1

                023554                  LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                023554                  LOC=-4&LOC
```

```
                    023560                          LOC=LOC+4
                    023560                          .=LOC
                    
023560  000240                  NB1:    NOP
023562  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
023564  005701                  NB2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
023566  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
023570  000240                          NOP
                                                                ;IF NONE OCCURS REPORT
023572  012737  010000  001230          MOV     #10000,$TMP2    ;ERROR!
023600  104127                  1$:     ERROR   127
023602  012737  177777  030760          MOV     #-1,MANFL2
023610  000530                  BR      NBDONE

023612                          NB3:

023612  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
023620  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
023626  012737  023702  000114          MOV     #NB6,@#CACHVEC  ;SET UP FOR ERROR
023634  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
023640  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
023644  000401                  BR      NB4

                    023646                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                    023644                          LOC=-4&LOC
                    023650                          LOC=LOC+4
                    023650                          .=LOC

023650  000240                  NB4:    NOP
023652  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
023654  005701                  NB5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
023656  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
023660  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
                                                                ;NO ABORT OCCURRED!
023662  012737  010000  001230          MOV     #10000,$TMP2    ;REPORT FAILURE
023670  104127                  1$:     ERROR   127
023672  012737  177777  030744          MOV     #-1,MANFLG
023700  000474                  BR      NBDONE

023702                          NB6:

023702  062706  000010                  ADD     #10,SP          ;RESET THE STACK.
023706  022737  137404  177744          CMP     #137404,@#MEMERR        ;SEE IF THE ERROR REGISTER
023714  001004                  BNE     NB7             ;IS SET CORRECTLY.
023716  022737  023564  177740          CMP     #NB2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
023724  001422                  BEQ     NB8             ;IS SET CORRECTLY.

023726                          NB7:                    ;NOT SET CORRECTLY!
023726  012737  137404  001230          MOV     #137404,$TMP2   ;REPORT FAILURE.
023734  013737  177744  001232          MOV     @#MEMERR,$TMP3
023742  012737  023564  001234          MOV     #NB2,$TMP4
023750  005037  001236                  CLR     $TMP5
023754  013737  177740  001240          MOV     @#LOADRS,$TMP6
023762  013737  177742  001242          MOV     @#HIADRS,$TMP7
```

```

023770  104135                          1$:     ERROR   135

023772  005037  177572                  NB8:    CLR     @#MMR0              ;TURN OFF MEMORY MANAGEMENT.
023776  005037  172516                          CLR     @#MMR3
024002  012737  177777  177744                  MOV     #-1,@#MEMERR        ;SEE IF YOU CAN CLR THE
024010  005737  177744                          TST     @#MEMERR            ;ERROR REG.
024014  001416                                  BEQ     NB10

024016                                  NB9:                                ;WON'T CLEAR!
024016  013737  177740  001230                  MOV     @#LOADRS,$TMP2
024024  013737  177742  001232                  MOV     @#HIADRS,$TMP3
024032  013737  177744  001234                  MOV     @#MEMERR,$TMP4

024040  104130                          1$:     ERROR   130
024042  012737  177777  030740                  MOV     #-1,MMRFLG
024050  000410                                  BR      NBDONE

024052  022737  177740  177740          NB10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
024060  001356                                  BNE     NB9                 ;HAS RESET
024062  022737  000003  177742                  CMP     #3,@#HIADRS
024070  001352                                  BNE     NB9

024072  104416                          NBDONE: RSET



                                        ;;*****************************************************************
                                        ;*TEST 51        CACHE ERROR REGISTER LOCK UP TEST 3
                                        ;*
                                        ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
                                        ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
                                        ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
                                        ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
                                        ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
                                        ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
                                        ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
                                        ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
                                        ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
                                        ;*TO THE CACHE DIRECTLY.
                                        ;*
                                        ;;*****************************************************************
024074  000004                          TST51:  SCOPE
024076  012737  000040  001274                  MOV     #40,$TIMES          ;;DO 40 ITERATIONS
        000051                          NC=$TN-1
                                                                            ;SET THE SKAD REGISTER
024104  012737  024574  030524                  MOV     #TST52,SKAD         ;IN CASE THE TEST ABORTS.

024112  113737  001102  001224                  MOVB    $TSTNM,$TMP0

024120  104430                                  SKPBER              ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
024122  104432                                  SKPBCN              ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
024124  104434                                  SKPBMN              ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
024126  104436                                  SKPBHM              ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
024130  104422                                  MMSKIP

024132  012700  172340                          MOV     #KIPAR0,R0          ;SET UP MEMORY MANAGEMENT
```

```
                                                                ;TO RELOCATE EVERYTHING
024136  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
024142  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
024146  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
024150  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                                ;RELOCATED TO THEMSELVES.
024154  012722  077406          64$:    MOV     #77406,(R2)+
024160  010401                          MOV     R4,R1
024162  072127  000006                  ASH     #6,R1
024166  010125                          MOV     R1,(R5)+
024170  005025                          CLR     (R5)+
024172  010410                          MOV     R4,(R0)
024174  062720  170000                  ADD     #170000,(R0)+
024200  062704  000200                  ADD     #200,R4
024204  077315                          SOB     R3,64$
024206  012710  177600                  MOV     #177600,(R0)
024212  012712  077406                  MOV     #77406,(R2)
024216  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
024224  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
024232  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
024240  012737  024316  000114          MOV     #NC3,@#CACHVEC  ;SET UP FOR ERROR
024246  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
024252  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
024256  000402                          BR      NC1
        024260                          LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        024260                          LOC=-4&LOC
        024264                          LOC=LOC+4
        024264                          .=LOC
024264  000240                  NC1:    NOP
024266  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
024270  005701                  NC2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
024272  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
024274  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
                                                                ;NO ABORT OCCURRED!
024276  012737  010000  001230          MOV     #10000,$TMP2    ;REPORT FAILURE
024304  104127                  1$:     ERROR   127
024306  012737  177777  030744          MOV     #-1,MANFLG
024314  000526                          BR      NCDONE
024316  005037  177572          NC3:    CLR     @#MMR0                  ;TURN OFF MEMORY MANAGEMENT.
024322  005037  172516                  CLR     @#MMR3
024326  012737  024402  000114          MOV     #NC6,@#CACHVEC          ;SET UP FOR THE ERROR.
024334  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
024340  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
024344  000401                          BR      NC4
        024346                          LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        024344                          LOC=-4&LOC
        024350                          LOC=LOC+4
```

```
6045            024350                          .=LOC
6046
6047  024350    000240                  NC4:    NOP
6048  024352    010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
6049  024354    005701                  NC5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
6050  024356    005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
6051  024360    000240                          NOP
6052                                                                    ;IF NONE OCCURS REPORT
6053  024362    012737  010000  001230          MOV     #10000,$TMP2    ;ERROR!
6054  024370    104127                  1$:     ERROR   127
6055  024372    012737  177777  030760          MOV     #-1,MANFL2
6056  024400    000474                  BR      NCDONE
6057
6058
6059  024402                            NC6:
6060
6061  024402    062706  000010                  ADD     #10,SP          ;RESET THE STACK.
6062  024406    022737  167404  177744          CMP     #167404,@#MEMERR        ;SEE IF THE ERROR REGISTER
6063  024414    001004                          BNE     NC7             ;IS SET CORRECTLY.
6064  024416    022737  024270  177740          CMP     #NC2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
6065  024424    001422                          BEQ     NC8             ;IS SET CORRECTLY.
6066
6067  024426                            NC7:                          ;NOT SET CORRECTLY!
6068  024426    012737  167404  001230          MOV     #167404,$TMP2   ;REPORT FAILURE.
6069  024434    013737  177744  001232          MOV     @#MEMERR,$TMP3
6070  024442    012737  024270  001234          MOV     #NC2,$TMP4
6071  024450    005037  001236                  CLR     $TMP5
6072  024454    013737  177740  001240          MOV     @#LOADRS,$TMP6
6073  024462    013737  177742  001242          MOV     @#HIADRS,$TMP7
6074
6075  024470    104135                  1$:     ERROR   135
6076
6077  024472    005037  177572          NC8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
6078  024476    005037  172516                  CLR     @#MMR3
6079  024502    012737  177777  177744          MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
6080  024510    005737  177744                  TST     @#MEMERR        ;ERROR REG.
6081  024514    001416                          BEQ     NC10
6082
6083  024516                            NC9:                          ;WON'T CLEAR!
6084  024516    013737  177740  001230          MOV     @#LOADRS,$TMP2
6085  024524    013737  177742  001232          MOV     @#HIADRS,$TMP3
6086  024532    013737  177744  001234          MOV     @#MEMERR,$TMP4
6087
6088  024540    104130                  1$:     ERROR   130
6089  024542    012737  177777  030740          MOV     #-1,MMRFLG
6090  024550    000410                          BR      NCDONE
6091
6092  024552    022737  177740  177740  NC10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
6093  024560    001356                          BNE     NC9             ;HAS RESET
6094  024562    022737  000003  177742          CMP     #3,@#HIADRS
6095  024570    001352                          BNE     NC9
6096
6097  024572    104416                  NCDONE: RSET
6098
6099
6100                                      ;;*****************************************************************
```

```
                                        ;*TEST 52       CACHE ERROR REGISTER LOCK UP TEST 4
                                        ;*
                                        ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
                                        ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
                                        ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
                                        ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
                                        ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
                                        ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
                                        ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
                                        ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
                                        ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
                                        ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
                                        ;*
                                        ;;*****************************************************************
024574  000004                  TST52:  SCOPE
024576  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000052                  ND=$TN-1
                                                                ;SET THE SKAD REGISTER
024604  012737  025300  030524          MOV     #TST53,SKAD     ;IN CASE THE TEST ABORTS.

024612  113737  001102  001224          MOVB    $TSTNM,$TMPO

024620  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
024622  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
024624  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
024626  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
024630  104422                          MMSKIP

024632  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                ;TO RELOCATE EVERYTHING
024636  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
024642  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
024646  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
024650  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                                ;RELOCATED TO THEMSELVES.
024654  012722  077406          64$:    MOV     #77406,(R2)+
024660  010401                          MOV     R4,R1
024662  072127  000006                  ASH     #6,R1
024666  010125                          MOV     R1,(R5)+
024670  005025                          CLR     (R5)+
024672  010410                          MOV     R4,(R0)
024674  062720  170000                  ADD     #170000,(R0)+
024700  062704  000200                  ADD     #200,R4
024704  077315                          SOB     R3,64$
024706  012710  177600                  MOV     #177600,(R0)
024712  012712  077406                  MOV     #77406,(R2)

024716  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.

024724  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
024732  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
024740  012737  025016  000114          MOV     #ND3,@#CACHVEC  ;SET UP FOR ERROR
024746  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
024752  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
024756  000402                          BR      ND1
```

```
6157
6158                024760                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
6159                024760                          LOC=-4&LOC
6160                024764                          LOC=LOC+4
6161                024764                          .=LOC
6162
6163  024764  000240                        ND1:    NOP
6164  024766  010412                              MOV     R4,(R2)         ;SET THE MAINT. REG.
6165  024770  005701                        ND2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
6166  024772  005012                              CLR     (R2)            ;SHOULD CASE AN ABORT
6167  024774  000240                              NOP                     ;AND UNIBUS PB ASSERTED!
6168                                                                      ;NO ABORT OCCURRED!
6169  024776  012737  010000  001230              MOV     #10000,$TMP2    ;REPORT FAILURE
6170  025004  104127                      1$:     ERROR   127
6171  025006  012737  177777  030744              MOV     #-1,MANFLG
6172  025014  000530                              BR      NDDONE
6173
6174
6175  025016                                ND3:
6176
6177  025016  012737  000060  172516              MOV     #60,@#MMR3      ;TURN ON THE MAP AND
6178  025024  012737  000001  177572              MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
6179  025032  012737  025106  000114              MOV     #ND6,@#CACHVEC  ;SET UP FOR ERROR
6180  025040  012704  010000                      MOV     #10000,R4       ;PATTERN TO BE PUT IN
6181  025044  012702  177750                      MOV     #MAINT,R2       ;THE MAINT. REG.
6182  025050  000401                              BR      ND4
6183
6184                025052                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
6185                025050                          LOC=-4&LOC
6186                025054                          LOC=LOC+4
6187                025054                          .=LOC
6188
6189  025054  000240                        ND4:    NOP
6190  025056  010412                              MOV     R4,(R2)         ;SET THE MAINT. REG.
6191  025060  005701                        ND5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
6192  025062  005012                              CLR     (R2)            ;SHOULD CASE AN ABORT
6193  025064  000240                              NOP                     ;AND UNIBUS PB ASSERTED!
6194                                                                      ;NO ABORT OCCURRED!
6195  025066  012737  010000  001230              MOV     #10000,$TMP2    ;REPORT FAILURE
6196  025074  104127                      1$:     ERROR   127
6197  025076  012737  177777  030744              MOV     #-1,MANFLG
6198  025104  000474                              BR      NDDONE
6199
6200
6201  025106                                ND6:
6202
6203  025106  062706  000010                      ADD     #10,SP          ;RESET THE STACK.
6204  025112  022737  033404  177744              CMP     #33404,@#MEMERR ;SEE IF THE ERROR REGISTER
6205  025120  001004                              BNE     ND7             ;IS SET CORRECTLY.
6206  025122  022737  024770  177740              CMP     #ND2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
6207  025130  001422                              BEQ     ND8             ;IS SET CORRECTLY.
6208
6209  025132                                ND7:                            ;NOT SET CORRECTLY!
6210  025132  012737  033404  001230              MOV     #33404,$TMP2    ;REPORT FAILURE.
6211  025140  013737  177744  001232              MOV     @#MEMERR,$TMP3
6212  025146  012737  024770  001234              MOV     #ND2,$TMP4
```

```
6213  025154  005037  001236                  CLR     $TMP5
6214  025160  013737  177740  001240          MOV     @#LOADRS,$TMP6
6215  025166  013737  177742  001242          MOV     @#HIADRS,$TMP7
6216
6217  025174  104135                  1$:     ERROR   135
6218
6219  025176  005037  177572          ND8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
6220  025202  005037  172516                  CLR     @#MMR3
6221  025206  012737  177777  177744          MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
6222  025214  005737  177744                  TST     @#MEMERR        ;ERROR REG.
6223  025220  001416                          BEQ     ND10
6224
6225  025222                          ND9:                            ;WON'T CLEAR!
6226  025222  013737  177740  001230          MOV     @#LOADRS,$TMP2
6227  025230  013737  177742  001232          MOV     @#HIADRS,$TMP3
6228  025236  013737  177744  001234          MOV     @#MEMERR,$TMP4
6229
6230  025244  104130                  1$:     ERROR   130
6231  025246  012737  177777  030740          MOV     #-1,MMRFLG
6232  025254  000410                          BR      NDDONE
6233
6234  025256  022737  177740  177740  ND10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
6235  025264  001356                          BNE     ND9             ;HAS RESET
6236  025266  022737  000003  177742          CMP     #3,@#HIADRS
6237  025274  001352                          BNE     ND9
6238
6239  025276  104416                  NDDONE: RSET
6240
6241
6242                                  ;;*****************************************************************
6243                                  ;*TEST 53       MAIN MEMORY DATA PARITY CHECKERS LOW BYTE TEST
6244                                  ;*
6245                                  ;*THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS
6246                                  ;*FOR THE LOW BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD.
6247                                  ;*THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY
6248                                  ;*ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY
6249                                  ;*BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE
6250                                  ;*THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT
6251                                  ;*A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE
6252                                  ;*AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY
6253                                  ;*BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS
6254                                  ;*SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1).
6255                                  ;*THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA
6256                                  ;*PARITY CHECKERS WORKS IN SUCH A WAY AS TO
6257                                  ;*EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO
6258                                  ;*THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO
6259                                  ;*AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS
6260                                  ;*ALREADY ONE THEN NO ERROR OCCURS!
6261                                  ;*
6262                                  ;*****************************************************************
6263  025300  000004          TST53:  SCOPE
6264  025302  012737  000020  001274          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
6265          000054          UA=$TN
6266                                                                  ;SET THE SKAD REGISTER
6267  025310  012737  025654  030524          MOV     #TST54,SKAD     ;IN CASE THE TEST ABORTS.
6268
```

```
6269  025316  113737  001102  001224          MOVB    $TSTNM,$TMP0
6270  025324  012737  030400  000114          MOV     #SPUR,@#CACHVEC
6271
6272  025332  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
6273  025340  005000                          CLR     R0              ;INITIALIZE
6274
6275  025342  012737  025342  001110  UA1:    MOV     #UA1,$LPERR
6276  025350  004737  030764                  JSR     PC,PARCNT       ;SEE IF THE CURRENT TEST
6277  025354  032702  000001                  BIT     #BIT0,R2        ;PATTERN HAS THE PARITY BIT
6278  025360  001002                          BNE     UA2             ;OFF.  IF NOT GO TO NEXT
6279  025362  000137  025634                  JMP     UA7             ;PATTERN
6280
6281  025366  012737  025540  000114  UA2:    MOV     #UAER1,@#CACHVEC        ;SET UP FOR THE ERROR, EVEN WORD.
6282  025374  012704  010000                  MOV     #10000,R4       ;THIS IS A PATTERN WHICH
6283  025400  012702  177750                  MOV     #MAINT,R2       ;WHEN LOADED INTO THE
6284                                                                  ;MAINTENANCE REGISTER
6285                                                                  ;WILL FORCE AN ERROR ON
6286                                                                  ;THE MAIN MEMORY EVEN
6287  025404  012701  025534                  MOV     #UATMP1,R1      ;WORD LOW BYTE
6288  025410  010011                          MOV     R0,(R1)
6289  025412  010412                          MOV     R4,(R2)         ;SET THE MAINT REG
6290  025414  021101                          CMP     (R1),R1         ;THE REFERENCE TO (R1),
6291                                                                  ;UATMP1 SHOULD CAUSE
6292                                                                  ;AN ERROR.
6293  025416  005012                          CLR     (R2)
6294  025420  005012                          CLR     (R2)
6295
6296  025422                          UA3:
6297                                                                  ;THE ERROR DIDN'T OCCUR!
6298  025422  010037  001230                  MOV     R0,$TMP2        ;REPORT FAILURE
6299  025426  012737  025534  001232          MOV     #UATMP1,$TMP3
6300  025434  005037  001234                  CLR     $TMP4
6301  025440  104140                  64$:    ERROR   140
6302
6303  025442  012737  025600  000114  UA4:    MOV     #UAER2,@#CACHVEC        ;SET UP FOR THE ERROR
6304  025450  012737  025442  001110          MOV     #UA4,$LPERR     ;ON THE ODD WORD.
6305  025456  012704  040000                  MOV     #40000,R4       ;THIS IS A PATTERN WHICH
6306  025462  012702  177750                  MOV     #MAINT,R2       ;WHEN LOADED IN THE MAINTENANCE
6307                                                                  ;REGISTER WILL CAUSE AN ERROR
6308  025466  012701  025536                  MOV     #UATMP2,R1      ;ON THE ODD WORD, LOW BYTE.
6309  025472  010011                          MOV     R0,(R1)         ;SET THE MAINT REG. AND
6310  025474  000240                          NOP
6311  025476  010412                          MOV     R4,(R2)         ;REFERENCE (R1), UATMP2, AND
6312  025500  021101                          CMP     (R1),R1         ;CAUSE THE ERROR.
6313
6314  025502  005012                          CLR     (R2)
6315  025504  005012                          CLR     (R2)
6316
6317  025506                          UA5:
6318                                                                  ;THE ERROR DIDN'T OCCUR!
6319  025506  010037  001230                  MOV     R0,$TMP2        ;REPORT FAILURE
6320  025512  012737  025536  001232          MOV     #UATMP2,$TMP3
6321  025520  005037  001234                  CLR     $TMP4
6322  025524  104141                  64$:    ERROR   141
6323
6324  025526  000442                  UA6:    BR      UA7
```

```
                        025530                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                        025530                          LOC=-4&LOC
                        025534                          LOC=LOC+4
                        025534                          .=LOC

025534  000000                    UATMP1:.WORD    0
025536  000000                    UATMP2:.WORD    0

025540                            UAER1:
025540  022737  104404  177744            CMP     #104404,@#MEMERR        ;MAKE SURE THE ERROR
025546  001402                            BEQ     2$                      ;REGISTER IS SET PROPERLY
025550  000137  030400            1$:     JMP     SPUR
025554  022737  025534  177740  2$:       CMP     #UATMP1,@#LOADRS        ;MAKE SURE THE ERROR
025562  001372                            BNE     1$                      ;OCCURRED AT THE CORRECT
                                                                          ;ADDRESS.
025564  022626                            CMP     (SP)+,(SP)+             ;RESET THE STACK
025566  012737  177777  177744            MOV     #-1,@#MEMERR            ;CLEAR THE ERROR REGISTERS.
025574  000137  025442                    JMP     UA4                     ;GO TEST THE ODD WORD

025600                            .AER2:
025600  022737  104410  177744            CMP     #104410,@#MEMERR        ;MAKE SURE THE ERROR
025606  001402                            BEQ     2$                      ;REGISTER IS SET PROPERLY
025610  000137  030400            1$:     JMP     SPUR
025614  022737  025536  177740  2$:       CMP     #UATMP2,@#LOADRS        ;MAKE SURE THE ERROR
025622  001372                            BNE     1$                      ;OCCURRED AT THE CORRECT
                                                                          ;ADDRESS.
025624  022626                            CMP     (SP)+,(SP)+             ;RESET THE STACK
025626  012737  177777  177744            MOV     #-1,@#MEMERR            ;CLEAR THE ERROR REGISTERS.

025634  022700  000377            UA7:    CMP     #377,R0                 ;INCREMENT THE TEST PATTERN
025640  001404                            BEQ     UA8
025642  062700  000001                    ADD     #1,R0
025646  000137  025342                    JMP     UA1

025652  104416                    UA8:    RSET

                                ;;*****************************************************************
                                ;*TEST 54        MAIN MEMORY DATA PARITY CHECKERS HIGH BYTE TEST
                                ;*
                                ;*THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS
                                ;*FOR THE HIGH BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD.
                                ;*THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY
                                ;*ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY
                                ;*BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE
                                ;*THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT
                                ;*A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE
                                ;*AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY
                                ;*BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS
                                ;*SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1).
                                ;*THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA
                                ;*PARITY CHECKERS WORKS IN SUCH A WAY AS TO
                                ;*EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO
                                ;*THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO
                                ;*AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS
```

```
                                        ;*ALREADY ONE THEN NO ERROR OCCURS!
                                        ;*
                                        ;;*****************************************************************
025654  000004                          TST54:  SCOPE
025656  012737  000020  001274                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000055                          UB=$TN
                                                                        ;SET THE SKAD REGISTER
025664  012737  026230  030524                  MOV     #TST55,SKAD     ;IN CASE THE TEST ABORTS.

025672  113737  001102  001224                  MOVB    $TSTNM,$TMPO
025700  012737  030400  000114                  MOV     #SPUR,@#CACHVEC

025706  012737  000014  177746                  MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
025714  005000                                  CLR     R0              ;INITIALIZE

025716  012737  025716  001110          UB1:    MOV     #UB1,$LPERR
025724  004737  030764                          JSR     PC,PARCNT       ;SEE IF THE CURRENT TEST
025730  032702  000001                          BIT     #BIT0,R2        ;PATTERN HAS THE PARITY BIT
025734  001002                                  BNE     UB2             ;OFF, IF NOT GO TO NEXT
025736  000137  026210                          JMP     UB7             ;PATTERN

025742  012737  026114  000114  UB2:    MOV     #UBER1,@#CACHVEC        ;SET UP FOR THE ERROR, EVEN WORD.
025750  012704  020000                          MOV     #20000,R4       ;THIS IS A PATTERN WHICH
025754  012702  177750                          MOV     #MAINT,R2       ;WHEN LOADED INTO THE
                                                                        ;MAINTENANCE REGISTER
                                                                        ;WILL FORCE AN ERROR ON
                                                                        ;THE MAIN MEMORY EVEN
025760  012701  026110                          MOV     #UBTMP1,R1      ;WORD HIGH BYTE
025764  010011                                  MOV     R0,(R1)
025766  010412                                  MOV     R4,(R2)         ;SET THE MAINT REG
025770  021101                                  CMP     (R1),R1         ;THE REFERENCE TO (R1),
                                                                        ;UBTMP1 SHOULD CAUSE
                                                                        ;AN ERROR.
025772  005012                                  CLR     (R2)
025774  005012                                  CLR     (R2)

025776                          UB3:
                                                                        ;THE ERROR DIDN'T OCCUR!
025776  010037  001230                          MOV     R0,$TMP2        ;REPORT FAILURE
026002  012737  026110  001232                  MOV     #UBTMP1,$TMP3
026010  005037  001234                          CLR     $TMP4
026014  104142                  64$:    ERROR   142

026016  012737  026154  000114  UB4:    MOV     #UBER2,@#CACHVEC        ;SET UP FOR THE ERROR
026024  012737  026016  001110                  MOV     #UB4,$LPERR     ;ON THE ODD WORD.
026032  012704  100000                          MOV     #100000,R4      ;THIS IS A PATTERN WHICH
026036  012702  177750                          MOV     #MAINT,R2       ;WHEN LOADED IN THE MAINTENANCE
                                                                        ;REGISTER WILL CAUSE AN ERROR
026042  012701  026112                          MOV     #UBTMP2,R1      ;ON THE ODD WORD, LOW BYTE.
026046  010011                                  MOV     R0,(R1)         ;SET THE MAINT REG. AND
026050  000240                                  NOP
026052  010412                                  MOV     R4,(R2)         ;REFERENCE (R1), UBTMP2, AND
026054  021101                                  CMP     (R1),R1         ;CAUSE THE ERROR.

026056  005012                                  CLR     (R2)
026060  005012                                  CLR     (R2)
```

```
                                UB5:
                                                                ;THE ERROR DIDN'T OCCUR!
                                        MOV     R0,$TMP2        ;REPORT FAILURE
                                        MOV     #UBTMP2,$TMP3
                                        CLR     $TMP4
                                64$:    ERROR   143

                                UB6:    SR      UB7

                                        LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                                        LOC=-4$LOC
                                        LOC=LOC+4
                                        .=LOC

                                UBTMP1:.WORD    0
                                UBTMP2:.WORD    0

                                UBER1:
                                        CMP     #104404,@#MEMERR        ;MAKE SURE THE ERROR
                                        BEQ     2$              ;REGISTER IS SET PROPERLY
                                1$:     JMP     SPUR
                                2$:     CMP     #UBTMP1,@#LOADRS        ;MAKE SURE THE ERROR
                                        BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
                                        CMP     (SP)+,(SP)+     ;RESET THE STACK
                                        MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
                                        JMP     UB4             ;GO TEST THE ODD WORD

                                UBER2:
                                        CMP     #104410,@#MEMERR        ;MAKE SURE THE ERROR
                                        BEQ     2$              ;REGISTER IS SET PROPERLY
                                1$:     JMP     SPUR
                                2$:     CMP     #UBTMP2,@#LOADRS        ;MAKE SURE THE ERROR
                                        BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
                                        CMP     (SP)+,(SP)+     ;RESET THE STACK
                                        MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.

                                UB7:    CMP     #177400,R0              ;INCREMENT THE TEST PATTERN
                                        BEQ     UB8
                                        ADD     #400,R0
                                        JMP     UB1

                                UB8:    RSET

                                TST55:

                                ;;*****************************************************************

                                .SBTTL  END OF PASS ROUTINE

                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
```

```
                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO LOOP
                                        $EOP:
                                                SCOPE
                                                CLR     $TSTNM          ;;ZERO THE TEST NUMBER
                                                CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
                                                INC     $PASS           ;;INCREMENT THE PASS NUMBER
                                                BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
                                                DEC     (PC)+           ;;LOOP?
                                        $EOPCT: .WORD   1
                                                BGT     $DOAGN          ;;YES
                                                MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
                                        $ENDCT: .WORD   1
                                                $EOPCT
                                                TYPE    ,$ENDMG         ;;TYPE "END PASS #"
                                                MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                                TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
                                        $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
                                                BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
                                                MOV     #125252,R3
                                                JSR     PC,CHAINQ
                                                MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
                                                BEQ     $DOAGN          ;;RETURN ADDRESS
                                                RESET                   ;;CLEAR THE WORLD
                                        $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
                                                NOP                     ;;SAVE ROOM
                                                NOP                     ;;FOR
                                                NOP                     ;;ACT11
                                        $DOAGN:
                                                JMP     @#LOOP          ;;RETURN
                                        $ENDMG: .ASCIZ  <15><12>/END PASS #/
                                        $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING

                                        ;;*****************************************************************

                                        .SBTTL  SCOPE HANDLER ROUTINE

                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW14=1         LOOP ON TEST
                                        ;*SW11=1         INHIBIT ITERATIONS
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*SW08=1         LOOP ON TEST IN SWR<6:0>
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT

                                        $SCOPE:
                                                ROL     @#SWR           ;;LOOP ON PRESENT TEST?
                                                BMI     $OVER           ;;YES IF SW14=1
```

```
6549                                    ;*****START OF CODE FOR THE XOR TESTER*****
6550  026376  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
6551                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
6552  026400  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
6553  026404  012737  026424  000004            MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
6554  026412  005737  177060                    TST     @#177060        ;;TIME OUT ON XOR?
6555  026416  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
6556  026422  000466                    BR      $SVLAD          ;;GO TO THE NEXT TEST
6557  026424  022626                    5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
6558  026426  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
6559  026432  000426                    BR      7$              ;;LOOP ON THE PRESENT TEST
6560  026434                            6$:;*****END OF CODE FOR THE XOR TESTER*****
6561  026434  032737  000400  177570            BIT     #BIT08,@#SWR    ;;LOOP ON SPEC. TEST?
6562  026442  001407                    BEQ     2$              ;;BR IF NO
6563  026444  013746  177570                    MOV     @#SWR,-(SP)     ;;SET DESIRED TEST NUM. FROM SWR
6564  026450  042716  000200                    BIC     #$SWRMK,(SP)    ;;STRIP AWAY UNDESIRED BITS
6565  026454  122637  001102                    CMPB    (SP)+,$TSTNM    ;;ON THE RIGHT TEST?
6566  026460  001462                    BEQ     $OVER           ;;BR IF YES
6567  026462  105737  001103            2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
6568  026466  001421                    BEQ     3$              ;;BR IF NO
6569  026470  123737  001115  001103            CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
6570  026476  101015                    BHI     3$              ;;BR IF NO
6571  026500  032737  001000  177570            BIT     #BIT09,@#SWR    ;;LOOP ON ERROR?
6572  026506  001404                    BEQ     4$              ;;BR IF NO
6573  026510  013737  001110  001106    7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
6574  026516  000443                    BR      $OVER
6575  026520  105037  001103            4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
6576  026524  005037  001274                    CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
6577  026530  000415                    BR      1$              ;;ESCAPE TO THE NEXT TEST
6578  026532  032737  004000  177570    3$:     BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
6579  026540  001011                    BNE     1$              ;;BR IF YES
6580  026542  005737  001100                    TST     $PASS           ;;IF FIRST PASS OF PROGRAM
6581  026546  001406                    BEQ     1$              ;;      INHIBIT ITERATIONS
6582  026550  005237  001104                    INC     $ICNT           ;;INCREMENT ITERATION COUNT
6583  026554  023737  001274  001104            CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
6584  026562  002021                    BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
6585  026564  012737  000001  001104    1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
6586  026572  013737  026642  001274            MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
6587  026600  105237  001102            $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
6588  026604  011637  001106                    MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
6589  026610  011637  001110                    MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
6590  026614  005037  001276                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
6591  026620  112737  000001  001115            MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
6592  026626  013737  001102  177570    $OVER:  MOV     $TSTNM,@#DISPLAY;;DISPLAY TEST NUMBER
6593  026634  013716  001106                    MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
6594  026640  000002                    RTI                             ;;FIXES PS
6595  026642  000001                    $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
6596
6597                                    ;;*****************************************************************
6598
6599                                    .SBTTL  ERROR HANDLER ROUTINE
6600
6601                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
6602                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
6603                                    ;*AND GO TO ERTYPE ON ERROR
6604                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
```

```
                                        ;*SW15=1        HALT ON ERROR
                                        ;*              HALT CAN OCCUR BEFORE AND AFTER THE ERROR TYPEOUT
                                        ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1        BELL ON ERROR
                                        ;*SW09=1        LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

026644                                  $ERROR:
026644  105237  001103                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
026650  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
026652  013737  001102  177570          MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
026660  005737  177570                  TST     @#SWR           ;;HALT ON ERROR = 1?
026664  100001                          BPL     8$              ;;BRANCH IF NO
026666  000000                          HALT                    ;;YES--HALT
026670  032737  002000  177570  8$:     BIT     #BIT10,@#SWR    ;;BELL ON ERROR?
026676  001402                          BEQ     1$              ;;NO - SKIP
026700  104400  001300                  TYPE    ,$BELL          ;;RING BELL
026704  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
026710  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
026714  162737  000002  001116          SUB     #2,$ERRPC
026722  117737  152170  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
026730  032737  020000  177570          BIT     #BIT13,@#SWR    ;;SKIP TYPEOUT IF SET
026736  001004                          BNE     2$              ;;SKIP TYPEOUTS
026740  004737  031230                  JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
026744  104400  001305                  TYPE    ,$CRLF
026750  005737  177570          2$:     TST     @#SWR           ;;HALT ON ERROR
026754  100001                          BPL     9$              ;;SKIP IF CONTINUE
026756  000000                          HALT                    ;;HALT ON ERROR!
026760  022737  026334  000042  9$:     CMP     #$ENDAD,42      ;;ACT-11?
026766  001001                          BNE     3$              ;;BRANCH IF NO
026770  000000                          HALT                    ;;YES
026772  032737  001000  177570  3$:     BIT     #BIT09,@#SWR    ;;LOOP ON ERROR SWITCH SET?
027000  001402                          BEQ     4$              ;;BR IF NO
027002  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
027006  005737  001276          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
027012  001402                          BEQ     5$              ;;BR IF NONE
027014  013716  001276                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
027020                          5$:
027020  012737  177777  177744          MOV     #-1,@#MEMERR
027026  005037  177766                  CLR     @#CPUERR
027032  000002                          RTI

                                        ;;**************************************************************

                                        .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                        ;*SAVE R0-R5
                                        ;*CALL:
                                        ;*      SAVREG
                                        ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                        ;*
                                        ;*TOP---(+16)
                                        ;* +2---(+18)
                                        ;* +4---R5
                                        ;* +6---R4
```

```
                                        ;* +8---R3
                                        ;*+10---R2
                                        ;*+12---R1
                                        ;*+14---R0
                                        
027034                                  $SAVREG:
027034  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
027036  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
027040  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
027042  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
027044  010446                                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
027046  010546                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
027050  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
027054  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
027060  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
027064  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
027070  000002                                 RTI
                                        
                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
027072                                  $RESREG:
027072  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
027076  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
027102  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
027106  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
027112  012605                                 MOV     (SP)+,R5        ;;POP STACK INTO R5
027114  012604                                 MOV     (SP)+,R4        ;;POP STACK INTO R4
027116  012603                                 MOV     (SP)+,R3        ;;POP STACK INTO R3
027120  012602                                 MOV     (SP)+,R2        ;;POP STACK INTO R2
027122  012601                                 MOV     (SP)+,R1        ;;POP STACK INTO R1
027124  012600                                 MOV     (SP)+,R0        ;;POP STACK INTO R0
027126  000002                                 RTI
                                        
                                        ;;*****************************************************************
                                        
                                        .SBTTL  TYPE ROUTINE
                                        
                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                        ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                        ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                        ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*
                                        ;*2) USING A JSR INSTRUCTION
                                        ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
                                        ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
                                        ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE
```

```
6717  027130  105737  001151          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
6718  027134  100002                          BPL     1$              ;;BR IF YES
6719  027136  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
6720  027140  000407                          BR      3$              ;;LEAVE
6721  027142  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
6722  027144  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
6723  027150  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
6724  027152  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
6725  027154  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
6726  027156  012600                          MOV     (SP)+,R0        ;;RESTORE R0
6727  027160  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
6728  027164  000002                          RTI                     ;;RETURN
6729  027166  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
6730  027172  001426                          BEQ     8$
6731  027174  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT
6732  027200  001004                          BNE     5$
6733  027202  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
6734  027204  104400  001305                  TYPE    $CRLF
6735  027210  000757                          BR      2$              ;;GET NEXT CHARACTER
6736  027212  004737  027274          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
6737  027216  123726  001150          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
6738  027222  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
6739  027224  013746  001146                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
6740                                                                  ;;AND THE NULL CHAR.
6741  027230  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
6742  027234  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
6743  027236  004737  027274                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
6744  027242  105337  027340                  DECB    $CHARCNT        ;;DON'T COUNT THE NULL AS A CHARACTER
6745  027246  000770                          BR      7$              ;;LOOP
6746
6747                                  ;;HORIZONTAL TAB PROCESSOR
6748
6749  027250  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
6750  027254  004737  027274          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
6751  027260  132737  000007  027340          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
6752  027266  001372                          BNE     9$              ;;TAB STOP
6753  027270  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
6754  027272  000726                          BR      2$              ;;GET NEXT CHARACTER
6755  027274  105777  151642          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
6756  027300  100375                          BPL     $TYPEC
6757  027302  116677  000002  151634          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6758  027310  122766  000015  000002          CMPB    #CR,2(SP)       ;;BRANCH IF
6759  027316  001003                          BNE     1$              ;;NOT <CR>
6760  027320  105037  027340                  CLRB    $CHARCNT        ;;
6761  027324  000406                          BR      $TYPEX          ;;EXIT
6762  027326  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;BRANCH IF
6763  027334  001402                          BEQ     $TYPEX          ;;<LF>
6764  027336  105227                          INCB    (PC)+           ;;INC SPACE
6765  027340  000000          $CHARCNT:.WORD  0               ;;COUNT
6766  027342  000207          $TYPEX: RTS     PC
6767
6768
6769                                  ;;**************************************************************
6770
6771                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
6772
```

```
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M               ;;M=1 OR 0
                                        ;*                                      ;;1=TYPE LEADING ZEROS
                                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT

027344  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
027350  116637  000001  027567          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
027356  112637  027571                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
027362  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
027366  000406                          BR      $TYPON
027370  112737  000001  027567  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
027376  112737  000006  027571          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
027404  112737  000005  027566  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
027412  010346                          MOV     R3,-(SP)        ;;SAVE R3
027414  010446                          MOV     R4,-(SP)        ;;SAVE R4
027416  010546                          MOV     R5,-(SP)        ;;SAVE R5
027420  113704  027571                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
027424  005404                          NEG     R4
027426  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
027432  110437  027570                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
027436  113704  027567                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
027442  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
027446  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
027450  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
027452  000404                          BR      3$              ;;GO DO MSB
027454  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
027456  006105                          ROL     R5
027460  006105                          ROL     R5
027462  010503                          MOV     R5,R3
027464  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
027466  105337  027570                  DECB    $OMODE          ;;TYPE THIS DIGIT?
027472  100016                          BPL     7$              ;;BR IF NO
027474  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
027500  001002                          BNE     4$              ;;TEST FOR 0
027502  005704                          TST     R4              ;;SUPPRESS THIS 0?
027504  001403                          BEQ     5$              ;;BR IF YES
027506  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
027510  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
027514  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
```

```
 6829  027520  110337  027564                    MOVB    R3,8$           ;;SAVE FOR TYPING
 6830  027524  104400  027564                    TYPE    8$              ;;GO TYPE THIS DIGIT
 6831  027530  105337  027566          7$:       DECB    $OCNT           ;;COUNT BY 1
 6832  027534  003347                            BGT     2$              ;;BR IF MORE TO DO
 6833  027536  002402                            BLT     6$              ;;BR IF DONE
 6834  027540  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 6835  027542  000744                            BR      2$              ;;GO DO THE LAST DIGIT
 6836  027544  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
 6837  027546  012604                            MOV     (SP)+,R4        ;;RESTORE R4
 6838  027550  012603                            MOV     (SP)+,R3        ;;RESTORE R3
 6839  027552  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 6840  027560  012616                            MOV     (SP)+,(SP)
 6841  027562  000002                            RTI                     ;;RETURN
 6842  027564     000                    8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 6843  027565     000                            .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 6844  027566     000                    $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 6845  027567     000                    $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 6846  027570  000000                    $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 6847
 6848                                    ;;**************************************************************
 6849
 6850                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 6851
 6852                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 6853                                    ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 6854                                    ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 6855                                    ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 6856                                    ;*REPLACED WITH SPACES.
 6857                                    ;*CALL:
 6858                                    ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 6859                                    ;*      TYPDS                   ;;GO TO THE ROUTINE
 6860
 6861  027572                            $TYPDS:
 6862  027572  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 6863  027574  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 6864  027576  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 6865  027600  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 6866  027602  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 6867  027604  012746  020200                    MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 6868  027610  016605  000020                    MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 6869  027614  100004                            BPL     1$              ;;BR IF INPUT IS POS.
 6870  027616  005405                            NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 6871  027620  112766  000055  000001            MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 6872  027626  005000                    1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 6873  027630  012703  030006                    MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 6874  027634  112723  000040                    MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 6875  027640  005002                    2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 6876  027642  016001  027776                    MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 6877  027646  160105                    3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 6878  027650  002402                            BLT     4$              ;;BR IF DONE
 6879  027652  005202                            INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 6880  027654  000774                            BR      3$
 6881  027656  060105                    4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 6882  027660  005702                            TST     R2              ;;CHECK IF BCD DIGIT=0
 6883  027662  001002                            BNE     5$              ;;FALL THROUGH IF 0
 6884  027664  105716                            TSTB    (SP)            ;;STILL DOING LEADING 0'S?
```

```
6885  027666  100407                          BMI     7$              ;;BR IF YES
6886  027670  106316                  5$:     ASLB    (SP)            ;;MSD?
6887  027672  103003                          BCC     6$              ;;BR IF NO
6888  027674  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
6889  027702  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
6890  027706  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
6891  027712  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
6892  027714  005720                          TST     (R0)+           ;;JUST INCREMENTING
6893  027716  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
6894  027722  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
6895  027724  003002                          BGT     8$              ;;GO TO EXIT
6896  027726  010502                          MOV     R5,R2           ;;GET THE LSD
6897  027730  000764                          BR      6$              ;;GO CHANGE TO ASCII
6898  027732  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
6899  027734  100003                          BPL     9$              ;;BR IF NO
6900  027736  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
6901  027744  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
6902  027746  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
6903  027750  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
6904  027752  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
6905  027754  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
6906  027756  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
6907  027760  104400  030006                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
6908  027764  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
6909  027772  012616                          MOV     (SP)+,(SP)
6910  027774  000002                          RTI                     ;;RETURN TO USER
6911  027776  023420                  $DTBL:  10000.
6912  030000  001750                          1000.
6913  030002  000144                          100.
6914  030004  000012                          10.
6915  030006  000004                  $DBLK:  .BLKW   4
6916
6917                                          ;;**************************************************************
6918
6919                                          .SBTTL  TRAP DECODER
6920
6921                                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
6922                                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
6923                                          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
6924                                          ;*GO TO THAT ROUTINE.
6925
6926  030016  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
6927  030020  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
6928  030024  005740                          TST     -(R0)           ;;BACKUP BY 2
6929  030026  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
6930  030030  016000  030036                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
6931  030034  000200                          RTS     R0              ;;GO TO ROUTINE
6932
6933
6934                                          .SBTTL  TRAP TABLE
6935
6936                                          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
6937                                          ;*BY THE "TRAP" INSTRUCTION.
6938
6939                                          ;       ROUTINE
6940                                          ;       -------
```

```
                                        $TRPAD:
                                                $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
                                                $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
                                                $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
                                                $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
                                                $TYPDS  ;;CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
                                                $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
                                                $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE

                                                CLEAN   ;;CALL=RSET     TRAP+16(104416) GO RESET ALL REGISTERS.
                                                ABORTT  ;;CALL=SKIPT    TRAP+20(104420) THIS WILL SKIP TO THE NEXT TEST
                                                MMDES   ;;CALL=MMSKIP   TRAP+22(104422) IF SWITCH # IS ON SKIP TO THE NEXT TEST
                                                MSIZER  ;;CALL=SIZE     TRAP+24(104424) DETERMINE THE HIGHEST ADDRESS IN MEMORY
                                                SKBADR  ;;CALL=SKPBAD   TRAP+26(104426) SKIP TEST IF ERROR ADDRESS REGISTER IS I
                                                SKBERR  ;;CALL=SKPBER   TRAP+30(104430) SKIP TEST IF ERROR REGISTER IS INOPERATI
                                                SKBCNR  ;;CALL=SKPBCN   TRAP+32(104432) SKIP TEST IF CONTROL REGISTER IS INOPERA
                                                SKBMNR  ;;CALL=SKPBMN   TRAP+34(104434) SKIP TEST IF MAINTENANCE REGISTER IS INO
                                                SKBHMR  ;;CALL=SKPBHM   TRAP+36(104436) SKIP TEST IF HIT/MISS REGISTER IS IN OPE

                                        ;;**************************************************************

                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;POWER DOWN ROUTINE
030076  012737  030224  000024 $PWRDN:  MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
030104  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
030112  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
030114  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
030116  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
030120  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
030122  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
030124  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
030126  010637  030230                MOV     SP,$SAVR6       ;;SAVE SP
030132  012737  030144  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
030140  000000                        HALT
030142  000776                        BR      .-2             ;;HANG UP

                                        ;POWER UP ROUTINE
030144  013706  030230 $PWRUP:  MOV     $SAVR6,SP       ;;GET SP
030150  005037  030230         CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
030154  005237  030230 1$:     INC     $SAVR6          ;;WAIT FOR THE INC
030160  001375                BNE     1$              ;;OF  WORD
030162  012605                MOV     (SP)+,R5        ;;POP STACK INTO R5
030164  012604                MOV     (SP)+,R4        ;;POP STACK INTO R4
030166  012603                MOV     (SP)+,R3        ;;POP STACK INTO R3
030170  012602                MOV     (SP)+,R2        ;;POP STACK INTO R2
030172  012601                MOV     (SP)+,R1        ;;POP STACK INTO R1
030174  012600                MOV     (SP)+,R0        ;;POP STACK INTO R0
030176  012737  030076  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
030204  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
030212  104400                        TYPE                    ;;REPORT THE POWER FAILURE
030214  032005 $PWRMG:  .WORD   POWERM          ;;POWER FAIL MESSAGE POINTER
030216  012716                MOV     (PC)+,(SP)      ;;RESTART AT START
030220  003010 $PWRAD:  .WORD   START          ;;RESTART ADDRESS
030222  000002                RTI
030224  000000 $ILLUP:  HALT                   ;;THE POWER UP SEQUENCE WAS STARTED
```

```
 6997  030226  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 6998  030230  000000              $SAVR6:   0                       ;;PUT THE SP HERE
 6999
 7000                                        ;;*****************************************************************
 7001
 7002                                        .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
 7003
 7004                                        ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
 7005                                        ;*UNSIGNED OCTAL ASCIZ NUMBER.
 7006                                        ;*CALL
 7007                                        ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 7008                                        ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
 7009                                        ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
 7010
 7011
 7012  030232  104412              $DB20:    SAVREG                  ;;SAVE ALL REGISTERS
 7013  030234  016601  000002              MOV     2(SP),R1          ;;PICKUP THE POINTER TO LOW WORD
 7014  030240  012705  030351              MOV     #$OCTVL+13.,R5    ;;POINTER TO DATA TABLE
 7015  030244  012704  000014              MOV     #12.,R4           ;;DO ELEVEN CHARACTERS
 7016  030250  012703  177770              MOV     #^C7,R3           ;;MASK
 7017  030254  012100                      MOV     (R1)+,R0          ;;LOWER WORD
 7018  030256  012101                      MOV     (R1)+,R1          ;;HIGH WORD
 7019  030260  005002                      CLR     R2                ;;TERMINATOR
 7020  030262  110245              1$:     MOVB    R2,-(R5)          ;;PUT CHARACTER IN DATA TABLE
 7021  030264  010002                      MOV     R0,R2             ;;GET THIS DIGIT
 7022  030266  005304                      DEC     R4                ;;COUNT THIS CHARACTER
 7023  030270  003007                      BGT     3$                ;;BR IF NOT THE LAST DIGIT
 7024  030272  001405                      BEQ     2$                ;;BR IF IT IS THE LAST DIGIT
 7025  030274  005205                      INC     R5                ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
 7026  030276  010566  000002              MOV     R5,2(SP)          ;;ASCIZ CHAR. & PUT IT ON THE STACK
 7027  030302  104414                      RESREG                    ;;RESTORE ALL REGISTERS
 7028  030304  000207                      RTS     PC                ;;RETURN TO USER
 7029  030306  006203              2$:     ASR     R3                ;;POSITION THE MASK FOR THE LAST DIGIT
 7030  030310  006001              3$:     ROR     R1                ;;POSITION THE BINARY NUMBER FOR
 7031  030312  006000                      ROR     R0                ;;      THE NEXT OCTAL DIGIT
 7032  030314  006001                      ROR     R1
 7033  030316  006000                      ROR     R0
 7034  030320  006001                      ROR     R1
 7035  030322  006000                      ROR     R0
 7036  030324  040302                      BIC     R3,R2             ;;MASK OUT ALL JUNK
 7037  030326  062702  000060              ADD     #'0,R2            ;;MAKE THIS CHAR. ASCII
 7038  030332  000753                      BR      1$                ;;GO PUT IT IN THE DATA TABLE
 7039  030334  000016              $OCTVL: .BLKB   14.               ;;RESERVE DATA TABLE
 7040
 7041                                        ;THIS ROUTINE IS CALLED BY UNEXPECTED TRAPS TO VECTOR ERRVEC.
 7042                                        ;THE ERROR IS REPORTED AND CONTROL IS TRANSFERRED BACK TO THE TEST
 7043                                        ;FOLLOWING THE ONE THAT WAS INTERRUPTED WHEN THE ERROR OCCURRED!
 7044  030352  011637  001226      CPSPUR: MOV     (SP),$TMP1
 7045  030356  012737  030374  001230      MOV     #1$,$TMP2
 7046  030364  013737  177766  001232      MOV     @#CPUERR,$TMP3
 7047  030372  022626                      CMP     (SP)+,(SP)+       ;RESET THE STACK
 7048  030374  104150              1$:     ERROR   150
 7049  030376  104420                      SKIPT
 7050
 7051                                        ;THIS ROUTINE HANDLE UNEXPECTED TRAPS TO #CACHVEC.
 7052  030400  012737  030470  000114  SPUR:   MOV     #10$,@#CACHVEC
```

```
7053  030406  013700  177744                  MOV     @#MEMERR,R0
7054  030412  032700  000014                  BIT     #14,R0          ;SEE IF IT WAS A MAIN MEMORY PARITY ERROR.
7055  030416  001403                          BEQ     9$
7056  030420  013700  177740                  MOV     @#LOADRS,R0     ;IF IT WAS THEN THE BAD PARITY IS
7057  030424  005710                          TST     (R0)            ;CACHED AND MUST BE PURGED!!!!!
7058  030426  012737  030400  000114  9$:     MOV     #SPUR,@#CACHVEC
7059  030434  013737  177744  001234          MOV     @#MEMERR,$TMP4  ;TRAP HERE IF AN UNEXPECTED
7060  030442  013737  177740  001226          MOV     @#LOADRS,$TMP1  ;ERROR, PARITY, OCCURS.
7061  030450  013737  177742  001230          MOV     @#HIADRS,$TMP2
7062  030456  011637  001232                  MOV     (SP),$TMP3
7063  030462  022626                          CMP     (SP)+,(SP)+
7064  030464  104014                  1$:     ERROR   14
7065  030466  104420                          SKIPT                   ;?????
7066  030470  022626                  10$:    CMP     (SP)+,(SP)+
7067  030472  000137  030426                  JMP     9$
7068
7069                                  ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL SKIPT.
7070                                  ;IT TELLS THE USER THAT THE CURRENT TEST HAS BEEN
7071                                  ;ABORTED AND THAT CONTROL IS BEING PASSED TO THE NEXT TEST.
7072  030476  011637  001226          ABORTT: MOV     (SP),$TMP1
7073  030502  112737  000015  001114          MOVB    #15,$ITEMB
7074  030510  022626                          CMP     (SP)+,(SP)+
7075  030512  004737  031230                  JSR     PC,ERTYPE
7076  030516  104416                          RSET
7077  030520  000177  000000                  JMP     @SKAD           ;GO TO @SKAD, WHICH SHOULD
7078                                                                  ;BE SET TO THE
7079  030524  000000                  SKAD:   .WORD   0               ;ADDRESS OF THE NEXT TEST.
7080
7081
7082                                  ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL RSET. IT CLEARS ALL
7083                                  ;THE IMPORTANE REGISTERS AND RESETS THE STACK.
7084  030526                          CLEAN:
7085
7086  030526  012737  030400  000114          MOV     #SPUR,@#CACHVEC
7087  030534  012737  030352  000004          MOV     #CPSPUR,@#ERRVEC
7088  030542  011637  030614                  MOV     (SP),BACKAD
7089  030546  012706  001100                  MOV     #STACK,SP
7090  030552  005037  177750                  CLR     @#MAINT         ;CLEAR ALL CONTROL AND ERROR
7091  030556  005037  177572                  CLR     @#MMR0          ;REGISTERS.
7092  030562  005037  172516                  CLR     @#MMR3
7093  030566  005037  177746                  CLR     @#CONTRL
7094  030572  012737  177777  177744          MOV     #-1,@#MEMERR
7095  030600  005037  177766                  CLR     @#CPUERR
7096  030604  005037  177776                  CLR     @#PSW
7097  030610  000177  000000                  JMP     @BACKAD
7098  030614  000000                  BACKAD: .WORD   0
7099
7100
7101                                  ;COME HERE TO TEST THE REGISTER FLAGS AND USE THEM TO DETERMINE WHETHER
7102                                  ;OR NOT TO SKIP A TEST WHICH RELIES ON THE FUNCTIONALLITY OF THAT REGISTER
7103                                  ;TO BE PROPERLY RUN.
7104                                  ;THESE ROUTINES ARE CALLED BY THE TRAP CATCHER CALLS:
7105                                  ;       SKPBAD          SKIPT IF BAD ERROR ADDRESS REGISTER
7106                                  ;       SKPBER          SKIPT IF BAD ERROR REGISTER
7107                                  ;       SKPBCN          SKIPT IF BAD CONTROL REGISTER
7108                                  ;       SKPBMN          SKIPT IF BAD MAINTENANCE REGISTER
```

```
                                        ;       SKPBHM          SKIPT IF BAD HIT/MISS REGISTER
                                        ;
                                        SKBADR: TST     LOAFLG
030616  005737  030734                          BNE     1$
030622  001004                                  TST     HIAFLG
030624  005737  030736                          BNE     1$
030630  001001                                  RTI
030632  000002                          1$:     TYPE
030634  104400                                  .WORD   ADRNG
030636  032767                                  BR      SKRNG
030640  000433
                                        SKBERR: TST     MMRFLG
030642  005737  030740                          BNE     1$
030646  001001                                  RTI
030650  000002                          1$:     TYPE
030652  104400                                  .WORD   ERRNG
030654  033077                                  BR      SKRNG
030656  000424
                                        SKBCNR: TST     CONFLG
030660  005737  030742                          BNE     1$
030664  001001                                  RTI
030666  000002                          1$:     TYPE
030670  104400                                  .WORD   CNRNG
030672  033177                                  BR      SKRNG
030674  000415
                                        SKBMNR: TST     MANFLG
030676  005737  030744                          BNE     1$
030702  001001                                  RTI
030704  000002                          1$:     TYPE
030706  104400                                  .WORD   MNRNG
030710  033301                                  BR      SKRNG
030712  000406
                                        SKBHMR: TST     HIMFLG
030714  005737  030746                          BNE     1$
030720  001001                                  RTI
030722  000002                          1$:     TYPE
030724  104400                                  .WORD   HMRNG
030726  033407
                                        SKRNG:  CMP     (SP)+,(SP)+             ;RESET THE STACK AND GO TO THE
030730  022626                                  SKIPT                           ;NEXT TEST!!!!!
030732  104420
                                        LOAFLG: .WORD   0                       ;THESE ARE FLAGS USED TO DESIGNATE
030734  000000                          HIAFLG: .WORD   0                       ;EITHER A GOOD OR A BAD REGISTER.
030736  000000                          MMRFLG: .WORD   0                       ;GOOD WILL BE DESIGNATED BY A
030740  000000                          CONFLG: .WORD   0                       ;0 BAD BY A NOT ZERO!!
030742  000000                          MANFLG: .WORD   0
030744  000000                          HIMFLG: .WORD   0
030746  000000                          LOAFL2: .WORD   0
030750  000000                          HIAFL2: .WORD   0
030752  000000                          MMRFL2: .WORD   0
030754  000000                          CONFL2: .WORD   0
030756  000000                          MANFL2: .WORD   0
030760  000000                          HIMFL2: .WORD   0
030762  000000
                                        ;THIS ROUTINE IS CALLED TO DETERMINE THE PARITY OF
```

```
                                        ;A DATA PATTERN. THE PATTERN WHICH IS TAKEN BY THIS
                                        ;ROUTINE AS ITS ARGUMENT SHOULD BE PUT IN R0. THEN
                                        ;TRANSFER CONTROL HERE BY EXECUTING:
                                        ;       JSR     PC,PARCNT
                                        ;WHEN THIS ROUTINE RETURNS THE NUMBER OF ON,(1), BITS
                                        ;IN R0 IS LEFT IN R2. THIS WOULD BE A NUMBER BETWEEN
                                        ;0 AND 16.
030764  012701  000001          PARCNT: MOV     #1,R1
030770  005002                          CLR     R2
030772  030100                  1$:     BIT     R1,R0
030774  001401                          BEQ     2$
030776  005202                          INC     R2
031000  006301                  2$:     ASL     R1
031002  103373                          BCC     1$
031004  000207                          RTS     PC

                                        ;THIS ROUTINE IS CALLED TO RESTORE THE TOP 1500 (DEC) WORDS IN THE
                                        ;FIRST 28K OF MEMORY. THIS SHOULD EFFECTIVELY RESTORE ANY MONITOR
                                        ;OR LOADER THAT WAS PRESENT BEFORE THIS PROGRAM BEGAN EXECUTION.
                                        ;CONTROL IS PASSED TO THIS ROUTINE BY AN INTERRUPT FROM THE TTY KEYBOARD
                                        ;WHEN ANY CHARACTER IS TYPED ON THE KEYBOARD. IF THE CHARACTER.
                                        ;TURNS OUT TO BE A ↑C (CONTROL-C) THEN MEMORY IS RESTORED. IF THE
                                        ;CHARACTER IS NOT ↑C THEN A RETURN IS MADE TO THE TEST FOLLOWING
                                        ;THE ONE WHOSE EXECUTION WAS INTERRUPTED BY THE KEYBOARD INTERRUPT.
031006  017700  150126          RESMON: MOV     @$TKB,R0
031012  104416                          RSET
031014  005003                          CLR     R3
031016  042700  000200                  BIC     #BIT7,R0        ;GET THE CHARACTER, INITIALIZE THE REGISTERS
031022  022700  000003                  CMP     #3,R0           ;AND SEE IF THE CHARACTER WAS ↑C.
031026  001027                          BNE     NOCNC           ;BRANCH AND GO TO NEXT TEST IF NOT.
031030  104400                          TYPE                    ;ECHOE THE CONTROL-C AS '↑C'
031032  031742                          .WORD   CONCMS
031034  012704  002734          CHAINQ: MOV     #↑D1500,R4              ;AND RESTORE THE MONITOR.
031040  012701  051166                  MOV     #BOTTOM+4,R1
031044  012702  160000                  MOV     #160000,R2
031050  012142                  1$:     MOV     (R1)+,-(R2)
031052  077402                          SOB     R4,1$
031054  012737  177777  031130        MOV     #-1,MONF        ;RESET THE MONITOR RESTORED FLAG.
031062  020327  125252                  CMP     R3,#125252      ;SEE IF THE MONITOR IS BEING RESTORED
                                                                ;BY THE .$EOP ROUTINE.
031066  001001                          BNE     STOP            ;IF NOT GO HALT, OTHERWISE RETURN TO .$EOP
031070  000207                          RTS     PC
031072  104400                  STOP:   TYPE                    ;TYPE THE MONITOR RESTORED MESSAGE.
031074  031746                          .WORD   MMESRS
031076  013737  031126  000060        MOV     MONTTY,@#TKVEC
031104  000000                          HALT                    ;AND HALT!!
031106  005077  150026          NOCNC:  CLR     @$TKB           ;NOT CONTROL C SO RETURN TO NEXT TEST.
031112  152777  000100  150016        BISB    #BIT6,@$TKS
031120  104416                          RSET
031122  000177  177376                JMP     @SKAD           ;RETURN.
031126  000000                  MONTTY: .WORD   0               ;TEMPORARY STORAGE FOR THE INITIAL
                                                                ;CONTENTS OF THE TTY KEYBOARD INTERRUPT VECTOR.
031130  177777                  MONF:   .WORD   177777          ;FLAG. IF NOT -1 THE MONITOR IS SAVED!!


                                        ;THIS ROUTINE IS CALLED BY THE TRAP CALL MMSKIP. IT LOOKS
```

```
                                        ;AT THE SWITCH REGISTER AND DETERMINES WHETHER OR NOT
                                        ;SWITCH #7 IS ON. IF SO THE CURRENT TEST IS SKIPPED
                                        ;AND THE NEXT TEST IS ENTERED. A SSKAD MUST BE ISSUED
                                        ;BEFORE THE MMSKIP.
                                        ;THE PURPOSE OF SWITCH #7 IS TO CAUSE THE DELETION OF THE
                                        ;EXECUTION OF ANY TEST WHICH RELIES ON MEMORY MANAGEMENT
                                        ;FOR ITS OPERATION.
                                        
031132  032737  000200  177570  MMDES:   BIT     #SW7,@#SWR
031140  001001                          BNE     1$              ;IS THE SWITCH ON?
031142  000002                          RTI                     ;NO, SO RETURN.
031144  022626                  1$:     CMP     (SP)+,(SP)+
031146  104416                          RSET
031150  000177  177350                  JMP     @SKAD           ;YES, GO TO THE NEXT TEST.
                                        ;THIS ROUTINE IS CALLED TO DETERMINE THE HIGHEST POSSIBLE
                                        ;ADDRESS IN MEMORY. IT IS CALLED THUS, BY TRAP CALL SIZE:
                                        ;               SIZE
                                        ;       LOORDA: .WORD   0
                                        ;       HIORDA: .WORD   0
                                        ;       NXTINST:
                                        ;THE LOW ORDER 16-BITS OF THE ADDRESS ARE LEFT IN THE
                                        ;WORD DIRECTLY FOLLOWING THE CALL. THE HIGH ORDER 6-BITS
                                        ;ARE LEFT IN THE NEXT WORD AND CONTROL IS RETURNED
                                        ;TO THE THIRD WORD FOLLOWING THE CALL.
031154  010046                  MSIZER: MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0 AND R1
031156  010146                          MOV     R1,-(SP)        ;GET THE ADDRESS OF
031160  016600  000004                  MOV     4(SP),R0        ;THE CALL OF THE STACK.
031164  013710  177760                  MOV     @#SIZELO,(R0)
031170  005060  000002                  CLR     2(R0)
031174  012701  000006                  MOV     #6,R1           ;ROTATE THE 16-BIT 'BLOCK'
                                                                ;NUMBER 6-BITS TO THE
031200  006310                  1$:     ASL     (R0)            ;LEFT AND TURN ON LOW ORDER
031202  006160  000002                  ROL     2(R0)           ;BITS 1-5 LEAVING BIT-0
031206  077104                          SOB     R1,1$           ;OFF SO AS TO CREATE
031210  052710  000076                  BIS     #76,(R0)        ;THE 22-BIT PHYSICAL ADDRESS OF
                                                                ;THE HIGHEST WORD IN
                                                                ;MEMORY.
031214  022020                          CMP     (R0)+,(R0)+     ;DETERMINE THE RETURN ADDRESS
031216  010066  000004                  MOV     R0,4(SP)        ;AND LEAVE ON THE STACK FOR
                                                                ;AN RTI.
031222  012601                          MOV     (SP)+,R1        ;RESTORE R1 AND R0.
031224  012600                          MOV     (SP)+,R0
031226  000002                          RTI                     ;RETURN
                                ;THIS ROUTINE IS USED TO TYPE AN ERROR MESSAGE
                                ;WHICH IS IN THE DATA TABLE. IT IS CALLED BY
                                ;THE $ERROR ROUTINE OR BY FIRST SETTING THE $ITEMB
                                ;BYTE EQUAL TO THE ERROR TABLE ITEM NUMBER THAT IS
                                ;TO BE PRINTED OUT AND THEN EXECUTING A JSR PC,ERTYPE
031230  104400                  ERTYPE: TYPE
031232  001305                          .WORD   $CRLF
031234  010046                          MOV     R0,-(SP)        ;SAVE R0
031236  005000                          CLR     R0
031240  113700  001114                  MOVB    $ITEMB,R0       ;GET THE ITEM NUMBER
031244  001005                          BNE     1$              ;ZERO?
031246  013746  001116                  MOV     $ERRPC,-(SP)    ;YES, TYPE JUST THE PC
031252  104402                          TYPOC                   ;OF THE ERROR CALL.
```

```
7277  031254  000137  031572              JMP     ERT5
7278
7279  031260  005300                1$:   DEC     R0              ;MAKE R0 AN INDEX FOR THE
7280  031262  072027  000003              ASH     #3,R0           ;ERROR TABLE
7281  031266  062700  001310              ADD     #$ERRTB,R0
7282  031272  012037  031302              MOV     (R0)+,2$        ;TYPE EM, ERROR MESSAGE.
7283  031276  001404                      BEQ     3$
7284  031300  104400                      TYPE
7285  031302  000000                2$:   .WORD   0
7286  031304  104400                      TYPE
7287  031306  001305                      .WORD   $CRLF
7288  031310  012037  031320        3$:   MOV     (R0)+,4$        ;TYPE DH, DATA HEADER
7289  031314  001404                      BEQ     5$
7290  031316  104400                      TYPE
7291  031320  000000                4$:   .WORD   0
7292  031322  104400                      TYPE
7293  031324  001305                      .WORD   $CRLF
7294  031326  010146                5$:   MOV     R1,-(SP)        ;SAVE R1
7295  031330  012001                      MOV     (R0)+,R1        ;GET DT, DATA TABLE ADDRESS
7296  031332  001002                      BNE     6$
7297  031334  000137  031570              JMP     ERT4            ;JMP IF NO ERROR TABLE.
7298  031340  012000                6$:   MOV     (R0)+,R0        ;GET DF, DATA FORMAT ADDRESS
7299  031342  105710                ERT1: TSTB    (R0)            ;DATA FORMAT ENTRY EQUALS
7300  031344  001003                      BNE     7$              ;ZERO?
7301  031346  013146                      MOV     @(R1)+,-(SP)    ;YES, SO TYPE A 16-BIT
7302  031350  104402                      TYPOC                   ;OCTAL NUMBER
7303  031352  000500                      BR      ERT2
7304  031354  122710  000001        7$:   CMPB    #1,(R0)         ;FORMAT EQUALS 1?
7305  031360  001003                      BNE     9$
7306  031362  013146                      MOV     @(R1)+,-(SP)    ;YES, TYPE A DECIMAL NUMBER
7307  031364  104410                      TYPDS
7308  031366  000472                      BR      ERT2
7309
7310  031370  122710  000002        9$:   CMPB    #2,(R0)         ;FORMAT 2?
7311  031374  001012                      BNE     9$
7312  031376  012146                85$:  MOV     (R1)+,-(SP)     ;YES, TYPE A 22-BIT NUMBR
7313  031400  004737  030232              JSR     PC,$DB2O        ;CALL $DB2O TO CONVERT THE
7314  031404  062716  000003              ADD     #3,(SP)         ;BINARY TO ASCII
7315  031410  012637  031416              MOV     (SP)+,29$       ;TYPE THE STRING
7316  031414  104400                      TYPE
7317  031416  000000                29$:  .WORD   0
7318  031420  000455                      BR      ERT2
7319
7320  031422  122710  000004        9$:   CMPB    #4,(R0)         ;FORMAT 4?
7321  031426  001004                      BNE     10$
7322  031430  013146                      MOV     @(R1)+,-(SP)    ;YES, TYPE A 16-BIT
7323  031432  104404                      TYPOS                   ;OCTAL NUMBER SUPRESSING
7324  031434     016                      .BYTE   16              ;LEADING ZEROES
7325  031435     000                      .BYTE   0
7326  031436  000446                      BR      ERT2
7327  031440  122710  000003        10$:  CMPB    #3,(R0)         ;FORMAT 3?
7328  031444  001007                      BNE     11$
7329  031446  013146                      MOV     @(R1)+,-(SP)    ;YES CONVERT 16-BIT
7330  031450  012737  177777  031576      MOV     #-1,TVADFL      ;VIRTUAL ADDRESS TO 32-BIT
7331  031456  004737  031604              JSR     PC,TYPVAD       ;PHYSICAL ADDRESS AND TYPE
7332  031462  000434                      BR      ERT2            ;RELOCATE ONLY IF SEG. IS ON!
```

```
7333  031464  122710  000005          11$:    CMPB    #5,(R0)         ;FORMAT 5?
7334  031470  001005                          BNE     12$
7335  031472  012137  031500                  MOV     (R1)+,20$       ;PRINT ASCIZ STRING
7336  031476  104400                          TYPE
7337  031500  000000          20$:    .WORD   0
7338  031502  000426                          BR      ERT3
7339
7340  031504  122710  000006          12$:    CMPB    #6,(R0)         ;FORMAT 6
7341  031510  001005                          BNE     13$
7342  031512  005037  031576                  CLR     TVADFL
7343  031516  004737  031604                  JSR     PC,TYPVAD
7344  031522  000414                          BR      ERT2
7345
7346  031524  122710  000007          13$:    CMPB    #7,(R0)         ;FORMAT 7?
7347  031530  001010                          BNE     14$
7348  031532  012146                          MOV     (R1)+,-(SP)
7349  031534  004737  030232                  JSR     PC,$DB2O
7350  031540  012637  031546                  MOV     (SP)+,45$
7351  031544  104400                          TYPE
7352  031546  000000          45$:    .WORD   0
7353  031550  000401                          BR      ERT2
7354
7355  031552  000000          14$:    HALT                            ;?????
7356
7357  031554  104400          ERT2:   TYPE                            ;PRINT A TAB AFTER TYPING AN
7358  031556  032052                  .WORD   $TAB                    ;ERROR TABLE ENTRY OF ALL MODES
7359                                                                  ;EXCEPT ASCIZ
7360  031560  005200          ERT3:   INC     R0                      ;POINT TO THE NEXT FORMAT BYTE
7361  031562  005711                  TST     (R1)                    ;IS THERE ANOTHER ENTRY?
7362  031564  001401                  BEQ     ERT4
7363  031566  000665                  BR      ERT1                    ;YES, PROCESS IT
7364                                                                  ;OTHERWISE:
7365  031570  012601          ERT4:   MOV     (SP)+,R1                ;RESTORE R1
7366  031572  012600          ERT5:   MOV     (SP)+,R0                ;RESTORE R0
7367  031574  000207                  RTS     PC                      ;AND RETURN
7368
7369  031576  000000          TVADFL: .WORD   0                       ;FLAG USED TO TELL TYVAD
7370                                                                  ;WHETHER TO CONDITIONALLY
7371                                                                  ;OR UNCONDITIONALLY RELOCATE
7372                                                                  ;WHEN TYPING AN ADDRESS,
7373                                                                  ;-1 OR 0 RESPECTIVELY
7374
7375  031600  000000          TVADLO: .WORD   0                       ;REGISTERS FOR THE 22-BIT
7376  031602  000000          TVADHI: .WORD   0                       ;ADDRESS COMPUTED BY TYVAD.
7377
7378                          ;ROUTINE WHICH CONVERTS A 16-BIT ADDRESS TO A 22-BIT
7379                          ;ADDRESS. IF TVADFL IS -1, THEN CONVERT TO THE 22-BIT
7380                          ;REAL ADDRESS DEPENDENT ON SEG BEING ON OR OFF FOR RELOCATION.
7381                          ;IF TVADFL IS ZERO THEN UNCONDITIONAL USE THE KERNAL
7382                          ;PAR WHICH IS APPROPIATE TO DO RELOCATION.
7383  031604  104412          TYPVAD: SAVREG
7384  031606  016601  000002          MOV     2(SP),R1                ;GET THE VIRTUAL
7385  031612  010137  031600          MOV     R1,TVADLO               ;ADDRESS
7386  031616  005037  031602          CLR     TVADHI
7387  031622  005737  031576          TST     TVADFL                  ;CONDITIONALLY RELOCATE?
7388  031626  001404                  BEQ     1$
```

```
7389  031630  032737  000001  177572          BIT     #1,@#MMR0       ;YES, SEE IF MEMORY
7390  031636  001424                          BEQ     2$              ;MANAGEMENT IS ON
7391  031640  005000                  1$:     CLR     R0              ;RELOCATE
7392  031642  073027  000003                  ASHC    #3,R0           ;LEFT SHIFT R0 AND R1
7393  031646  006300                          ASL     R0              ;THREE PLACES. R0 ONE
7394                                                                  ;MORE SO THAT IT CONTAINS
7395                                                                  ;2 X THE UPPER 3-BITS OF
7396  031650  000241                          CLC                     ;THE VIRTUAL ADDRESS
7397  031652  006001                          ROR     R1              ;RESTORE R1 TO THE OFFSET
7398  031654  006001                          ROR     R1              ;OF THE VIRTUAL ADDRESS
7399  031656  006001                          ROR     R1              ;TO THE PAR
7400  031660  062700  172340                  ADD     #KIPAR0,R0      ;DETERMINE THE CORRECT PAR'S
7401                                                                  ;ADDRESS
7402  031664  011003                          MOV     (R0),R3         ;GET ITS CONTENTS
7403  031666  005002                          CLR     R2
7404  031670  073227  000006                  ASHC    #6,R2           ;MAKE THE BLOCK COUNT
7405                                                                  ;A 22-BIT ADDRESS.
7406  031674  060103                          ADD     R1,R3           ;ADD THE OFFSET TO THE
7407  031676  005502                          ADC     R2              ;BASE ADDRESS
7408
7409  031700  010237  031602                  MOV     R2,TVADHI
7410  031704  010337  031600                  MOV     R3,TVADLO
7411  031710  012746  031600          2$:     MOV     #TVADLO,-(SP)   ;CALL $DB2O TO CONVERT THE
7412  031714  004737  030232                  JSR     PC,$DB2O        ;22-BIT
7413  031720  062716  000003                  ADD     #3,(SP)         ;TYPE ONLY 9 DIGITS.
7414  031724  012637  031732                  MOV     (SP)+,3$
7415  031730  104400                          TYPE
7416  031732  000000                  3$:     .WORD   0
7417  031734  104414                          RESREG                  ;RESTORE THE REGISTERS
7418  031736  012616                          MOV     (SP)+,(SP)      ;LEAVE ONLY THE RETURN
7419                                                                  ;ADDRESS ON THE STACK.
7420  031740  000207                          RTS     PC              ;RETURN
7421
7422                                  ;SPECIAL MESSAGES:
7423
7424  031742  041536  000200          CONCMS: .ASCIZ  '↑C'<CRLF>
7425
7426  031746  047515  044516  047524  MMESRS: .ASCIZ  'MONITOR (OR LOADER) RESTORED!'<CRLF>
7427  031754  020122  047450  020122
7428  031762  047514  042101  051105
7429  031770  020051  042522  052123
7430  031776  051117  042105  100041
7431  032004     000
7432
7433  032005     200  047520  042527  POWERM: .ASCIZ  <CRLF>'POWER FAILURE, PROGRAM RESTARTING'<CRLF><CRLF>
7434  032012  020122  040506  046111
7435  032020  051125  026105  050040
7436  032026  047522  051107  046501
7437  032034  051040  051505  040524
7438  032042  052122  047111  100107
7439  032050  000200
7440
7441  032052  000011          $TAB:   .ASCIZ  <TAB>
7442
7443  032054  042600  050130  041505  MTA5:   .ASCII  <CRLF>'EXPECTED DATA:'<CRLF>
7444  032062  042524  020104  040504
```

```
 7445  032070  040524  100072
 7446  032074  051107  052517  020120          .ASCIZ  'GROUP 0.GROUP 1.MEM EV.'<TAB>'MEM ODD.'<CRLF>
 7447  032102  027060  051107  052517
 7448  032110  020120  027061  042515
 7449  032116  020115  053105  004456
 7450  032124  042515  020115  042117
 7451  032132  027104  000200
 7452
 7453  032136  042200  052101  020101  MTA11:  .ASCII  <CRLF>'DATA WRITTEN.'<TAB>'TEST ADDR.'<TAB>'ERROR REG.'<CRLF>
 7454  032144  051127  052111  042524
 7455  032152  027116  052011  051505
 7456  032160  020124  042101  051104
 7457  032166  004456  051105  047522
 7458  032174  020122  042522  027107
 7459  032202     200
 7460
 7461  032203     040  047111  000040  MTA17:  .ASCIZ  ' IN '
 7462
 7463  032210  054105  042520  052103  MTB17:  .ASCIZ  'EXPECTED DATA:'<CRLF>
 7464  032216  042105  042040  052101
 7465  032224  035101  000200
 7466
 7467  032230  054502  042524  004456  MTC17:  .ASCIZ  'BYTE.'<TAB>
 7468  032236     000
 7469
 7470  032237     127  051117  027104  MTA20:  .ASCIZ  'WORD.'<TAB>
 7471  032244  000011
 7472
 7473  032246  054105  042520  052103  MTA21:  .ASCII  'EXPECTED DATA:'<CRLF>
 7474  032254  042105  042040  052101
 7475  032262  035101     200
 7476  032265     110  052111  020123          .ASCIZ  'HITS IN GROUP 0.'<TAB>'/'<TAB>'HITS IN GROUP 1. '<CRLF>
 7477  032272  047111  043440  047522
 7478  032300  050125  030040  004456
 7479  032306  004457  044510  051524
 7480  032314  044440  020116  051107
 7481  032322  052517  020120  027061
 7482  032330  100040     000
 7483
 7484          032203                  MTB21=MTA17
 7485
 7486  032333     200  042524  052123  MTA43:  .ASCII  <CRLF>'TEST ADDRESS.'<TAB>'ERROR ADRS REG.'<TAB>
 7487  032340  040440  042104  042522
 7488  032346  051523  004456  051105
 7489  032354  047522  020122  042101
 7490  032362  051522  051040  043505
 7491  032370  004456
 7492  032372  051105  047522  020122          .ASCIZ  'ERROR REG.'<CRLF>
 7493  032400  042522  027107  000200
 7494
 7495  032406  053600  047522  042524  MTA45:  .ASCIZ  <CRLF>'WROTE. 377'<TAB>'IN BYTE. '
 7496  032414  020056  033463  004467
 7497  032422  047111  041040  052131
 7498  032430  027105  000040
 7499
 7500  032434  051200  040505  020104  MTB45:  .ASCIZ  <CRLF>'READ DATA. '
```

```
7501  032442  040504  040524  020056
7502  032450     000
7503
7504  032451     011  047111  053440  MTC45:  .ASCIZ  <TAB>'IN WORD. '
7505  032456  051117  027104  000040
7506
7507  032464  053600  047522  042524  MTA50:  .ASCIZ  <CRLF>'WROTE. 000'<TAB>'IN BYTE. '
7508  032472  020056  030060  004460
7509  032500  047111  041040  052131
7510  032506  027105  000040
7511
7512  032512  042600  052116  051105  PDMSG1: .ASCII  <CRLF>'ENTERING CACHE ADDRESS MEMORY POWER UP '
7513  032520  047111  020107  040503
7514  032526  044103  020105  042101
7515  032534  051104  051505  020123
7516  032542  042515  047515  054522
7517  032550  050040  053517  051105
7518  032556  052440  020120
7519  032562  047111  040526  044514          .ASCII  'INVALIDATOR TEST.'<CRLF>
7520  032570  040504  047524  020122
7521  032576  042524  052123  100056
7522  032604  046120  040505  042523          .ASCII  'PLEASE GO THROUGH A POWER DOWN, POWER UP '
7523  032612  043440  020117  044124
7524  032620  047522  043525  020110
7525  032626  020101  047520  042527
7526  032634  020122  047504  047127
7527  032642  020054  047520  042527
7528  032650  020122  050125     040
7529  032655     123  050505  042525          .ASCIZ  'SEQUENCE.'<CRLF>
7530  032662  041516  027105  000200
7531
7532  032670  041600  041501  042510  PDMSG2: .ASCII  <CRLF>'CACHE ADDRESS MEMORY POWER UP INVALIDATOR'
7533  032676  040440  042104  042522
7534  032704  051523  046440  046505
7535  032712  051117  020131  047520
7536  032720  042527  020122  050125
7537  032726  044440  053116  046101
7538  032734  042111  052101  051117
7539  032742  052040  051505  020124          .ASCIZ  ' TEST DID NOT FAIL.'<CRLF>
7540  032750  044504  020104  047516
7541  032756  020124  040506  046111
7542  032764  100056     000
7543
7544  032767     105  051122  051117  ADRNG:  .ASCII  'ERROR ADDRESS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
7545  032774  040440  042104  042522
7546  033002  051523  051040  043505
7547  033010  051511  042524  020122
7548  033016  042516  042105  042105
7549  033024  043040  051117  052040
7550  033032  051505  026124  041200
7551  033040  052125  044440  020124
7552  033046  040510  020123  042502
7553  033054  047105     040
7554  033057     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
7555  033064  042105  040440  020123
7556  033072  040502  020504     000
```

```
7557
7558  033077    105  051122  051117  ERRNG:  .ASCII  'ERROR REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
7559  033104 051040  043505  051511
7560  033112 042524  020122  042516
7561  033120 042105  042105  043040
7562  033126 051117  052040  051505
7563  033134 026124  041200  052125
7564  033142 044440  020124  040510
7565  033150 020123  042502  047105
7566  033156    040
7567  033157    106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
7568  033164 042105  040440  020123
7569  033172 040502  020504     000
7570
7571  033177    103  047117  051124  CNRNG:  .ASCII  'CONTROL REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
7572  033204 046117  051040  043505
7573  033212 051511  042524  020122
7574  033220 042516  042105  042105
7575  033226 043040  051117  052040
7576  033234 051505  026124  041200
7577  033242 052125  044440  020124
7578  033250 040510  020123  042502
7579  033256 047105     040
7580  033261    106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
7581  033266 042105  040440  020123
7582  033274 040502  020504     000
7583  033301    115  044501  052116  MNRNG:  .ASCII  'MAINTENANCE REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
7584  033306 047105  047101  042503
7585  033314 051040  043505  051511
7586  033322 042524  020122  042516
7587  033330 042105  042105  043040
7588  033336 051117  052040  051505
7589  033344 026124  041200  052125
7590  033352 044440  020124  040510
7591  033360 020123  042502  047105
7592  033366    040
7593  033367    106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
7594  033374 042105  040440  020123
7595  033402 040502  020504     000
7596
7597  033407    110  052111  046457  HMRNG:  .ASCII  'HIT/MISS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
7598  033414 051511  020123  042522
7599  033422 044507  052123  051105
7600  033430 047040  042505  042504
7601  033436 020104  047506  020122
7602  033444 042524  052123  100054
7603  033452 052502  020124  052111
7604  033460 044040  051501  041040
7605  033466 042505  020116
7606  033472 046106  043501  042507          .ASCIZ  'FLAGGED AS BAD!'
7607  033500 020104  051501  041040
7608  033506 042101  000041
7609
7610  033512 040600  042104  042522  MTA77:  .ASCIZ  <CRLF>'ADDRESS:  '
7611  033520 051523  020072  000040
7612
```

```
7613  033526  051440  047510  046125  MTB77:  .ASCIZ  ' SHOULD HAVE BEEN A HIT IN GROUP '
7614  033534  020104  040510  042526
7615  033542  041040  042505  020116
7616  033550  020101  044510  020124
7617  033556  047111  043440  047522
7618  033564  050125  000040
7619
7620  033570  043101  042524  020122  MTC77:  .ASCIZ  'AFTER REFERENCING'<CRLF>'ADDRESS:  '
7621  033576  042522  042506  042522
7622  033604  041516  047111  100107
7623  033612  042101  051104  051505
7624  033620  035123  020040     000
7625
7626  033625     040  044127  046111  MTD77:  .ASCIZ  ' WHILE FORCING SELECTION OF GROUP '
7627  033632  020105  047506  041522
7628  033640  047111  020107  042523
7629  033646  042514  052103  047511
7630  033654  020116  043117  043440
7631  033662  047522  050125  000040
7632
7633  033670  040600  051122  051117  MTA101: .ASCII  <CRLF>'ARROR ADRS REG.'<TAB>'ERROR REG.'<TAB>
7634  033676  040440  051104  020123
7635  033704  042522  027107  042411
7636  033712  051122  051117  051040
7637  033720  043505  004456
7638  033724  054105  042520  052103          .ASCIZ  'EXPECTED ERR.'<TAB>'PATTERN PUT IN MAINT REG.'<CRLF>
7639  033732  042105  042440  051122
7640  033740  004456  040520  052124
7641  033746  051105  020116  052520
7642  033754  020124  047111  046440
7643  033762  044501  052116  051040
7644  033770  043505  100056     000
7645
7646  033775     200  043101  042524  MTA120: .ASCIZ  <CRLF>'AFTER 2ND CYCLE READ  '
7647  034002  020122  047062  020104
7648  034010  054503  046103  020105
7649  034016  042522  042101  020040
7650  034024     000
7651
7652  034025     200  043101  042524  MTB120: .ASCIZ  <CRLF>'AFTER 4TH CYCLE READ  '
7653  034032  020122  052064  020110
7654  034040  054503  046103  020105
7655  034046  042522  042101  020040
7656  034054     000
7657
7658  034055     200  043101  042524  MTC120: .ASCIZ  <CRLF>'AFTER 6TH CYCLE READ  '
7659  034062  020122  052066  020110
7660  034070  054503  046103  020105
7661  034076  042522  042101  020040
7662  034104     000
7663  034105     200  043101  042524  MTD120: .ASCIZ  <CRLF>'AFTER 8TH CYCLE READ  '
7664  034112  020122  052070  020110
7665  034120  054503  046103  020105
7666  034126  042522  042101  020040
7667  034134     000
7668
```

```
7669  034135    200  043101  042524  MTE120: .ASCIZ  <CRLF>'AFTER 10TH CYCLE READ '
7670  034142 020122  030061  044124
7671  034150 041440  041531  042514
7672  034156 051040  040505  020104
7673  034164    000
7674
7675  034165    200  043101  042524  MTF120: .ASCIZ  <CRLF>'AFTER 12TH CYCLE READ '
7676  034172 020122  031061  044124
7677  034200 041440  041531  042514
7678  034206 051040  040505  020104
7679  034214    000
7680
7681  034215    106  047522  020115  MTG120: .ASCIZ  'FROM THE HIT/MISS REG. EXPECTED '
7682  034222 044124  020105  044510
7683  034230 027524  044515  051523
7684  034236 051040  043505  020056
7685  034244 054105  042520  052103
7686  034252 042105  000040
7687
7688  034256 052200  042510  050040  MTA124: .ASCII  <CRLF>'THE PATTERN BEING USED IN THE MAINTENANCE '
7689  034264 052101  042524  047122
7690  034272 041040  044505  043516
7691  034300 052440  042523  020104
7692  034306 047111  052040  042510
7693  034314 046440  044501  052116
7694  034322 047105  047101  042503
7695  034330    040
7696  034331    122  043505  051511          .ASCIZ  'REGISTER WAS: '
7697  034336 042524  020122  040527
7698  034344 035123  000040
7699
7700  034350 051200  043105  051105  MTA126: .ASCIZ  <CRLF>'REFERENCED ADDRESS:'<TAB>
7701  034356 047105  042503  020104
7702  034364 042101  051104  051505
7703  034372 035123  000011
7704
7705  034376 040600  051122  051117  MTB126: .ASCIZ  <CRLF>'ARROR ADDRESS REGISTER:'<TAB>
7706  034404 040440  042104  042522
7707  034412 051523  051040  043505
7708  034420 051511  042524  035122
7709  034426 000011
7710
7711  034430 050200  052101  042524  MTA131: .ASCIZ  <CRLF>'PATTERN BEING USED IN THE MAINTENANCE REGISTER:'<TAB>
7712  034436 047122  041040  044505
7713  034444 043516  052440  042523
7714  034452 020104  047111  052040
7715  034460 042510  046440  044501
7716  034466 052116  047105  047101
7717  034474 042503  051040  043505
7718  034502 051511  042524  035122
7719  034510 000011
7720
7721  034512 042600  050130  041505  MTB131: .ASCIZ  <CRLF>'EXPECTED ERROR REGISTER:'<TAB>
7722  034520 042524  020104  051105
7723  034526 047522  020122  042522
7724  034534 044507  052123  051105
```

```
7725  034542  004472      000
7726
7727  034545      200  047507  020124  MTC131: .ASCIZ  <CRLF>'GOT ERROR REGISTER:'<TAB>
7728  034552  051105  047522  020122
7729  034560  042522  044507  052123
7730  034566  051105  004472      000
7731
7732  034573      200  051105  047522  MTA134: .ASCIZ  <CRLF>'ERROR ADR REG.'<TAB>'ERROR REG.'<CRLF>
7733  034600  020122  042101  020122
7734  034606  042522  027107  042411
7735  034614  051122  051117  051040
7736  034622  043505  100056      000
7737
7738  034627      200  054105  042520  MTA135: .ASCIZ  <CRLF>'EXPECTED ERROR REG.:  '
7739  034634  052103  042105  042440
7740  034642  051122  051117  051040
7741  034650  043505  035056  020040
7742  034656      000
7743
7744  034657      107  052117  042440  MTB135: .ASCIZ  'GOT ERROR REG.:  '
7745  034664  051122  051117  051040
7746  034672  043505  035056  020040
7747  034700      000
7748
7749  034701      200  054105  042520  MTC135: .ASCIZ  <CRLF>'EXPECTED ERROR ADR REG.:  '
7750  034706  052103  042105  042440
7751  034714  051122  051117  040440
7752  034722  051104  051040  043505
7753  034730  035056  020040      000
7754
7755  034735      107  052117  042440  MTD135: .ASCIZ  'GOT ERROR ADR REG.:  '
7756  034742  051122  051117  040440
7757  034750  051104  051040  043505
7758  034756  035056  020040      000
7759
7760
7761                                  ;THESE ARE THE ERROR MESSAGES:
7762
7763  034763      101  051040  043105  EM1:    .ASCIZ  'A REFERENCE WHICH SHOULD HAVE BEEN A HIT WAS A MISS.'
7764  034770  051105  047105  042503
7765  034776  053440  044510  044103
7766  035004  051440  047510  046125
7767  035012  020104  040510  042526
7768  035020  041040  042505  020116
7769  035026  020101  044510  020124
7770  035034  040527  020123  020101
7771  035042  044515  051523  000056
7772
7773
7774  035050  052600  042516  050130  EM14:   .ASCIZ  <CRLF>'UNEXPECTED PARITY ERROR TRAP.'
7775  035056  041505  042524  020104
7776  035064  040520  044522  054524
7777  035072  042440  051122  051117
7778  035100  052040  040522  027120
7779  035106      000
7780
```

```
 7781  035107     052  025052  042524  EM15:   .ASCIZ  '***TEST ABORTED! GOING TO NEXT TEST.***'
 7782  035114  052123  040440  047502
 7783  035122  052122  042105  020041
 7784  035130  047507  047111  020107
 7785  035136  047524  047040  054105
 7786  035144  020124  042524  052123
 7787  035152  025056  025052     000
 7788  035157     103  041501  042510  EM55:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 7789  035164  051040  043505  051511
 7790  035172  042524  020122  042522
 7791  035200  050123  047117  042523
 7792  035206  052040  051505  020124
 7793  035214  040506  046111  042105
 7794  035222  100056
 7795  035224  020101  042522  042506          .ASCII  'A REFERENCE TO THE LOW ORDER ERROR ADDRESS REGISTER '
 7796  035232  042522  041516  020105
 7797  035240  047524  052040  042510
 7798  035246  046040  053517  047440
 7799  035254  042122  051105  042440
 7800  035262  051122  051117  040440
 7801  035270  042104  042522  051523
 7802  035276  051040  043505  051511
 7803  035304  042524  020122
 7804  035310  044524  042515  020104          .ASCIZ  'TIMED OUT.'
 7805  035316  052517  027124     000
 7806
 7807  035323     103  041501  042510  EM56:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 7808  035330  051040  043505  051511
 7809  035336  042524  020122  042522
 7810  035344  050123  047117  042523
 7811  035352  052040  051505  020124
 7812  035360  040506  046111  042105
 7813  035366  100056
 7814  035370  020101  042522  042506          .ASCII  'A REFERENCE TO THE HIGH ORDER ERROR ADDRESS REGISTER '
 7815  035376  042522  041516  020105
 7816  035404  047524  052040  042510
 7817  035412  044040  043511  020110
 7818  035420  051117  042504  020122
 7819  035426  051105  047522  020122
 7820  035434  042101  051104  051505
 7821  035442  020123  042522  044507
 7822  035450  052123  051105     040
 7823  035455     124  046511  042105          .ASCIZ  'TIMED OUT.'
 7824  035462  047440  052125  000056
 7825
 7826  035470  040503  044103  020105  EM57:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 7827  035476  042522  044507  052123
 7828  035504  051105  051040  051505
 7829  035512  047520  051516  020105
 7830  035520  042524  052123  043040
 7831  035526  044501  042514  027104
 7832  035534     200
 7833  035535     101  051040  043105          .ASCIZ  'A REFERENCE TO THE ERROR REGISTER TIMED OUT.'
 7834  035542  051105  047105  042503
 7835  035550  052040  020117  044124
 7836  035556  020105  051105  047522
```

```
 7837  035564  020122  042522  044507
 7838  035572  052123  051105  052040
 7839  035600  046511  042105  047440
 7840  035606  052125  000056
 7841
 7842  035612  040503  044103  020105  EM60:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 7843  035620  042522  044507  052123
 7844  035626  051105  051040  051505
 7845  035634  047520  051516  020105
 7846  035642  042524  052123  043040
 7847  035650  044501  042514  027104
 7848  035656     200
 7849  035657     101  051040  043105          .ASCIZ  'A REFERENCE TO THE CONTROL REGISTER TIMED OUT.'
 7850  035664  051105  047105  042503
 7851  035672  052040  020117  044124
 7852  035700  020105  047503  052116
 7853  035706  047522  020114  042522
 7854  035714  044507  052123  051105
 7855  035722  052040  046511  042105
 7856  035730  047440  052125  000056
 7857
 7858  035736  040503  044103  020105  EM61:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 7859  035744  042522  044507  052123
 7860  035752  051105  051040  051505
 7861  035760  047520  051516  020105
 7862  035766  042524  052123  043040
 7863  035774  044501  042514  027104
 7864  036002     200
 7865  036003     101  051040  043105          .ASCIZ  'A REFERENCE TO THE MAINTENANCE REGISTER TIMED OUT.'
 7866  036010  051105  047105  042503
 7867  036016  052040  020117  044124
 7868  036024  020105  040515  047111
 7869  036032  042524  040516  041516
 7870  036040  020105  042522  044507
 7871  036046  052123  051105  052040
 7872  036054  046511  042105  047440
 7873  036062  052125  000056
 7874
 7875  036066  040503  044103  020105  EM62:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 7876  036074  042522  044507  052123
 7877  036102  051105  051040  051505
 7878  036110  047520  051516  020105
 7879  036116  042524  052123  043040
 7880  036124  044501  042514  027104
 7881  036132     200
 7882  036133     101  051040  043105          .ASCIZ  'A REFERENCE TO THE HIT/MISS REGISTER TIMED OUT.'<CRLF>
 7883  036140  051105  047105  042503
 7884  036146  052040  020117  044124
 7885  036154  020105  044510  027524
 7886  036162  044515  051523  051040
 7887  036170  043505  051511  042524
 7888  036176  020122  044524  042515
 7889  036204  020104  052517  027124
 7890  036212  000200
 7891
 7892  036214  040503  044103  020105  EM63:   .ASCII  'CACHE REGISTER DATA PATHS, READ ZEROES, TEST FAILED.'
```

```
 7893  036222  042522  044507  052123
 7894  036230  051105  042040  052101
 7895  036236  020101  040520  044124
 7896  036244  026123  051040  040505
 7897  036252  020104  042532  047522
 7898  036260  051505  020054  042524
 7899  036266  052123  043040  044501
 7900  036274  042514  027104
 7901  036300  053600  047522  042524          .ASCII  <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA '
 7902  036306  055040  051105  042517
 7903  036314  020123  052502  020124
 7904  036322  042522  042101  041040
 7905  036330  041501  020113  047516
 7906  036336  026516  042532  047522
 7907  036344  042040  052101  020101
 7908  036352  051106  046517  041040          .ASCIZ  'FROM BOTH'<CRLF>'THE CONTROL AND MAINTENANCE REGISTERS.'
 7909  036360  052117  100110  044124
 7910  036366  020105  047503  052116
 7911  036374  047522  020114  047101
 7912  036402  020104  040515  047111
 7913  036410  042524  040516  041516
 7914  036416  020105  042522  044507
 7915  036424  052123  051105  027123
 7916  036432     000
 7917
 7918  036433     103  041501  042510  EM64:   .ASCII  'CACHE REGISTER DATA PATH, READ ZEROES, TEST FAILED.'
 7919  036440  051040  043505  051511
 7920  036446  042524  020122  040504
 7921  036454  040524  050040  052101
 7922  036462  026110  051040  040505
 7923  036470  020104  042532  047522
 7924  036476  051505  020054  042524
 7925  036504  052123  043040  044501
 7926  036512  042514  027104
 7927  036516  053600  047522  042524          .ASCII  <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA FROM '
 7928  036524  055040  051105  042517
 7929  036532  020123  052502  020124
 7930  036540  042522  042101  041040
 7931  036546  041501  020113  047516
 7932  036554  026516  042532  047522
 7933  036562  042040  052101  020101
 7934  036570  051106  046517     040
 7935  036575     200  044124  020105          .ASCIZ  <CRLF>'THE MAINTENANCE REGISTER.'
 7936  036602  040515  047111  042524
 7937  036610  040516  041516  020105
 7938  036616  042522  044507  052123
 7939  036624  051105  000056
 7940
 7941  036630  040503  044103  020105  EM65:   .ASCII  'CACHE REGISTER DATA PATHS, READ ONES, REST FAILED.'<CRLF>
 7942  036636  042522  044507  052123
 7943  036644  051105  042040  052101
 7944  036652  020101  040520  044124
 7945  036660  026123  051040  040505
 7946  036666  020104  047117  051505
 7947  036674  020054  042522  052123
 7948  036702  043040  044501  042514
```

```
 7949  036710  027104     200
 7950  036713     106  044501  042514          .ASCII  'FAILED TO READ CORRECT DATA FROM THE ADDRESS REGISTER'
 7951  036720  020104  047524  051040
 7952  036726  040505  020104  047503
 7953  036734  051122  041505  020124
 7954  036742  040504  040524  043040
 7955  036750  047522  020115  044124
 7956  036756  020105  042101  051104
 7957  036764  051505  020123  042522
 7958  036772  044507  052123  051105
 7959  037000  044440  020116  044124          .ASCII  ' IN THE CLEAR STATE.'<CRLF>'THE LOW ORDER ADDRESS '
 7960  037006  020105  046103  040505
 7961  037014  020122  052123  052101
 7962  037022  027105  052200  042510
 7963  037030  046040  053517  047440
 7964  037036  042122  051105  040440
 7965  037044  042104  042522  051523
 7966  037052     040
 7967  037053     123  047510  046125          .ASCII  'SHOULD HAVE BEEN SET TO: 177740'<CRLF>
 7968  037060  020104  040510  042526
 7969  037066  041040  042505  020116
 7970  037074  042523  020124  047524
 7971  037102  020072  033461  033467
 7972  037110  030064     200
 7973  037113     124  042510  044040          .ASCII  'THE HIGH ORDER ADDRESS REGISTER SHOULD HAVE BEEN '
 7974  037120  043511  020110  051117
 7975  037126  042504  020122  042101
 7976  037134  051104  051505  020123
 7977  037142  042522  044507  052123
 7978  037150  051105  051440  047510
 7979  037156  046125  020104  040510
 7980  037164  042526  041040  042505
 7981  037172  020116
 7982  037174  042523  020124  047524          .ASCIZ  'SET TO: 000003'
 7983  037202  020072  030060  030060
 7984  037210  031460     000
 7985
 7986  037213     103  041501  042510  EM66:   .ASCIZ  'CACHE CONTROL REGISTER COUNT PATTERN TEST FAILED.'
 7987  037220  041440  047117  051124
 7988  037226  046117  051040  043505
 7989  037234  051511  042524  020122
 7990  037242  047503  047125  020124
 7991  037250  040520  052124  051105
 7992  037256  020116  042524  052123
 7993  037264  043040  044501  042514
 7994  037272  027104     000
 7995
 7996  037275     103  041501  042510  EM67:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 7997  037302  044040  052111  046457
 7998  037310  051511  020123  047101
 7999  037316  020104  047503  052116
 8000  037324  047522  020114  042522
 8001  037332  044507  052123  051105
 8002  037340  052040  051505  020124
 8003  037346  040506  046111  042105
 8004  037354     056
```

```
 8005  037355    200  044527  044124          .ASCII  <CRLF>'WITH THE CONTROL REGISTER CLEAR, THE HIT/MISS '
 8006  037362 052040  042510  041440
 8007  037370 047117  051124  046117
 8008  037376 051040  043505  051511
 8009  037404 042524  020122  046103
 8010  037412 040505  026122  052040
 8011  037420 042510  044040  052111
 8012  037426 046457  051511  020123
 8013  037434 042522  044507  052123          .ASCIZ  'REGISTER SHOULD'<CRLF>'HAVE SHOWN SIX HITS (000077).'
 8014  037442 051105  051440  047510
 8015  037450 046125  100104  040510
 8016  037456 042526  051440  047510
 8017  037464 047127  051440  054111
 8018  037472 044040  052111  020123
 8019  037500 030050  030060  033460
 8020  037506 024467  000056
 8021
 8022  037512 040503  044103  020105  EM70:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 8023  037520 044510  027524  044515
 8024  037526 051523  040440  042116
 8025  037534 041440  047117  051124
 8026  037542 046117  051040  043505
 8027  037550 051511  042524  020122
 8028  037556 042524  052123  043040
 8029  037564 044501  042514  027104
 8030  037572 053600  044510  042514          .ASCII  <CRLF>'WHILE FORCING SELECTION OF GROUP 1 AND FORCING '
 8031  037600 043040  051117  044503
 8032  037606 043516  051440  046105
 8033  037614 041505  044524  047117
 8034  037622 047440  020106  051107
 8035  037630 052517  020120  020061
 8036  037636 047101  020104  047506
 8037  037644 041522  047111  020107
 8038  037652 044515  051523  051505          .ASCII  'MISSES TO GROUP 0,'<CRLF>'THE HIT/MISS REGISTER '
 8039  037660 052040  020117  051107
 8040  037666 052517  020120  026060
 8041  037674 052200  042510  044040
 8042  037702 052111  046457  051511
 8043  037710 020123  042522  044507
 8044  037716 052123  051105     040
 8045  037723    123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX HITS (000077).'
 8046  037730 020104  040510  042526
 8047  037736 051440  047510  047127
 8048  037744 051440  054111  044040
 8049  037752 052111  020123  030050
 8050  037760 030060  033460  024467
 8051  037766 000056
 8052
 8053  037770 040503  044103  020105  EM71:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 8054  037776 044510  027524  044515
 8055  040004 051523  040440  042116
 8056  040012 041440  047117  051124
 8057  040020 046117  051040  043505
 8058  040026 051511  042524  020122
 8059  040034 042524  052123  043040
 8060  040042 044501  042514  027104
```

```
 8061  040050  053600  044510  042514          .ASCII  <CRLF>'WHILE FORCING SELECTION OF GROUP 0 AND FORCING '
 8062  040056  043040  051117  044503
 8063  040064  043516  051440  046105
 8064  040072  041505  044524  047117
 8065  040100  047440  020106  051107
 8066  040106  052517  020120  020060
 8067  040114  047101  020104  047506
 8068  040122  041522  047111  020107
 8069  040130  044515  051523  051505          .ASCII  'MISSES TO GROUP 1.'<CRLF>'THE HIT/MISS REGISTER '
 8070  040136  052040  020117  051107
 8071  040144  052517  020120  026061
 8072  040152  052200  042510  044040
 8073  040160  052111  046457  051511
 8074  040166  020123  042522  044507
 8075  040174  052123  051105     040
 8076  040201     123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX HITS (000077).'
 8077  040206  020104  040510  042526
 8078  040214  051440  047510  047127
 8079  040222  051440  054111  044040
 8080  040230  052111  020123  030050
 8081  040236  030060  033460  024467
 8082  040244  000056
 8083
 8084  040246  040503  044103  020105  EM72:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 8085  040254  044510  027524  044515
 8086  040262  051523  040440  042116
 8087  040270  041440  047117  051124
 8088  040276  046117  051040  043505
 8089  040304  051511  042524  020122
 8090  040312  042524  052123  043040
 8091  040320  044501  042514  027104
 8092  040326  044127  046111  020105          .ASCII  'WHILE FORCING MISSES TO BOTH GROUPS, THE HIT/MISS '
 8093  040334  047506  041522  047111
 8094  040342  020107  044515  051523
 8095  040350  051505  052040  020117
 8096  040356  047502  044124  043440
 8097  040364  047522  050125  026123
 8098  040372  052040  042510  044040
 8099  040400  052111  046457  051511
 8100  040406  020123
 8101  040410  042522  044507  052123          .ASCIZ  'REGISTER'<CRLF>'SHOULD HAVE SHOWN SIX MISSES (000000).'
 8102  040416  051105  051600  047510
 8103  040424  046125  020104  040510
 8104  040432  042526  051440  047510
 8105  040440  047127  051440  054111
 8106  040446  046440  051511  042523
 8107  040454  020123  030050  030060
 8108  040462  030060  024460  000056
 8109
 8110  040470  040503  044103  020105  EM73:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 8111  040476  044510  027524  044515
 8112  040504  051523  040440  042116
 8113  040512  041440  047117  051124
 8114  040520  046117  051040  043505
 8115  040526  051511  042524  020122
 8116  040534  042524  052123  043040
```

```
 8117  040542  044501  042514  027104
 8118  040550  053600  044510  042514          .ASCII  <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '
 8119  040556  043040  051117  044503
 8120  040564  043516  046440  051511
 8121  040572  042523  020123  047524
 8122  040600  041040  052117  020110
 8123  040606  051107  052517  051520
 8124  040614  040440  042116  043040
 8125  040622  051117  044503  043516
 8126  040630     040
 8127  040631     123  046105  041505          .ASCII  'SELECTION OF GROUP 1.'<CRLF>'THE HIT/MISS REGISTER '
 8128  040636  044524  047117  047440
 8129  040644  020106  051107  052517
 8130  040652  020120  026061  052200
 8131  040660  042510  044040  052111
 8132  040666  046457  051511  020123
 8133  040674  042522  044507  052123
 8134  040702  051105     040
 8135  040705     123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX MISSES (000000).'
 8136  040712  020104  040510  042526
 8137  040720  051440  047510  047127
 8138  040726  051440  054111  046440
 8139  040734  051511  042523  020123
 8140  040742  030050  030060  030060
 8141  040750  024460  000056
 8142
 8143  040754  040503  044103  020105  EM74:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 8144  040762  044510  027524  044515
 8145  040770  051523  040440  042116
 8146  040776  041440  047117  051124
 8147  041004  046117  051040  043505
 8148  041012  051511  042524  020122
 8149  041020  042524  052123  043040
 8150  041026  044501  042514  027104
 8151  041034  053600  044510  042514          .ASCII  <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '
 8152  041042  043040  051117  044503
 8153  041050  043516  046440  051511
 8154  041056  042523  020123  047524
 8155  041064  041040  052117  020110
 8156  041072  051107  052517  051520
 8157  041100  040440  042116  043040
 8158  041106  051117  044503  043516
 8159  041114     040
 8160  041115     123  046105  041505          .ASCII  'SELECTION OF GROUP 0.'<CRLF>'THE HIT/MISS REGISTER '
 8161  041122  044524  047117  047440
 8162  041130  020106  051107  052517
 8163  041136  020120  026060  052200
 8164  041144  042510  044040  052111
 8165  041152  046457  051511  020123
 8166  041160  042522  044507  052123
 8167  041166  051105     040
 8168  041171     123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX MISSES (000000).'
 8169  041176  020104  040510  042526
 8170  041204  051440  047510  047127
 8171  041212  051440  054111  046440
 8172  041220  051511  042523  020123
```

```
 8173  041226  030050  030060  030060
 8174  041234  024460  000056
 8175
 8176  041240  047503  052116  047522  EM75:   .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>'FAILED TO GET '
 8177  041246  020114  042522  044507
 8178  041254  052123  051105  052040
 8179  041262  051505  020124  040506
 8180  041270  046111  042105  100056
 8181  041276  040506  046111  042105
 8182  041304  052040  020117  042507
 8183  041312  020124
 8184  041314  020101  044510  020124          .ASCIZ  'A HIT ON A REFERENCE WHICH SHOULD HAVE BEEN A HIT.'
 8185  041322  047117  040440  051040
 8186  041330  043105  051105  047105
 8187  041336  042503  053440  044510
 8188  041344  044103  051440  047510
 8189  041352  046125  020104  040510
 8190  041360  042526  041040  042505
 8191  041366  020116  020101  044510
 8192  041374  027124     000
 8193
 8194          041240                  EM76=EM75
 8195
 8196  041377     103  047117  051124  EM77:   .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>'THE WRONG '
 8197  041404  046117  051040  043505
 8198  041412  051511  042524  020122
 8199  041420  042524  052123  043040
 8200  041426  044501  042514  027104
 8201  041434  052200  042510  053440
 8202  041442  047522  043516     040
 8203  041447     107  047522  050125          .ASCIZ  'GROUP WAS WRITTEN WHILE FORCING SELECTION OF A GROUP.'
 8204  041454  053440  051501  053440
 8205  041462  044522  052124  047105
 8206  041470  053440  044510  042514
 8207  041476  043040  051117  044503
 8208  041504  043516  051440  046105
 8209  041512  041505  044524  047117
 8210  041520  047440  020106  020101
 8211  041526  051107  052517  027120
 8212  041534     000
 8213
 8214  041535     103  047117  051124  EM117:  .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>
 8215  041542  046117  051040  043505
 8216  041550  051511  042524  020122
 8217  041556  042524  052123  043040
 8218  041564  044501  042514  027104
 8219  041572     200
 8220  041573     107  052117  040440          .ASCIZ  'GOT A HIT IN THE GROUP TO WHICH MISSES ARE BEING FORCED.'
 8221  041600  044040  052111  044440
 8222  041606  020116  044124  020105
 8223  041614  051107  052517  020120
 8224  041622  047524  053440  044510
 8225  041630  044103  046440  051511
 8226  041636  042523  020123  051101
 8227  041644  020105  042502  047111
 8228  041652  020107  047506  041522
```

```
 8229  041660  042105  000056
 8230
 8231  041664  044510  027524  044515  EM120:  .ASCII  'HIT/MISS REGISTER PATTERNS TEST FAILED.'
 8232  041672  051523  051040  043505
 8233  041700  051511  042524  020122
 8234  041706  040520  052124  051105
 8235  041714  051516  052040  051505
 8236  041722  020124  040506  046111
 8237  041730  042105     056
 8238  041733     200  042522  042101          .ASCII  <CRLF>'READ WRONG DATA FROM THE HIT/MISS REGISTER'<CRLF>
 8239  041740  053440  047522  043516
 8240  041746  042040  052101  020101
 8241  041754  051106  046517  052040
 8242  041762  042510  044040  052111
 8243  041770  046457  051511  020123
 8244  041776  042522  044507  052123
 8245  042004  051105     200
 8246  042007     127  044510  042514          .ASCIZ  'WHILE FLOATING A PATTERN OF HITS AND MISSES THROUGH IT.'
 8247  042014  043040  047514  052101
 8248  042022  047111  020107  020101
 8249  042030  040520  052124  051105
 8250  042036  020116  043117  044040
 8251  042044  052111  020123  047101
 8252  042052  020104  044515  051523
 8253  042060  051505  052040  051110
 8254  042066  052517  044107  044440
 8255  042074  027124     000
 8256
 8257  042077     103  041501  042510  EM121:  .ASCII  /CACHE CONTROL SIGNAL, THE 'RANDOM' SIGNAL, TEST FAILED./
 8258  042104  041440  047117  051124
 8259  042112  046117  051440  043511
 8260  042120  040516  026114  052040
 8261  042126  042510  023440  040522
 8262  042134  042116  046517  020047
 8263  042142  044523  047107  046101
 8264  042150  020054  042524  052123
 8265  042156  043040  044501  042514
 8266  042164  027104
 8267  042166  043200  044501  042514          .ASCIZ  <CRLF>'FAILED TO GET BOTH HITS AT THE TWO TEST ADDRESSES '
 8268  042174  020104  047524  043440
 8269  042202  052105  041040  052117
 8270  042210  020110  044510  051524
 8271  042216  040440  020124  044124
 8272  042224  020105  053524  020117
 8273  042232  042524  052123  040440
 8274  042240  042104  042522  051523
 8275  042246  051505     040
 8276  042251     127  044510  044103          .ASCIZ  'WHICH WERE REFERENCED.'
 8277  042256  053440  051105  020105
 8278  042264  042522  042506  042522
 8279  042272  041516  042105  000056
 8280
 8281  042300  040515  047111  042524  EM122:  .ASCII  'MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'
 8282  042306  040516  041516  020105
 8283  042314  042522  044507  052123
 8284  042322  051105  041440  052517
```

```
 8285  042330  052116  050040  052101
 8286  042336  042524  047122  052040
 8287  042344  051505  020124  040506
 8288  042352  046111  042105     056
 8289  042357     200  044124  020105          .ASCII  <CRLF>'THE MAINTENANCE REGISTER WILL NOT CLEAR.'
 8290  042364  040515  047111  042524
 8291  042372  040516  041516  020105
 8292  042400  042522  044507  052123
 8293  042406  051105  053440  046111
 8294  042414  020114  047516  020124
 8295  042422  046103  040505  027122
 8296
 8297  042430  040503  044103  020105  EM123:  .ASCII  'CACHE MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'
 8298  042436  040515  047111  042524
 8299  042444  040516  041516  020105
 8300  042452  042522  044507  052123
 8301  042460  051105  041440  052517
 8302  042466  052116  050040  052101
 8303  042474  042524  047122  052040
 8304  042502  051505  020124  040506
 8305  042510  046111  042105     056
 8306  042515     200  043101  042524          .ASCII  <CRLF>'AFTER WRITING A PATTERN IN THIS REGISTER '
 8307  042522  020122  051127  052111
 8308  042530  047111  020107  020101
 8309  042536  040520  052124  051105
 8310  042544  020116  047111  052040
 8311  042552  044510  020123  042522
 8312  042560  044507  052123  051105
 8313  042566     040
 8314  042567     106  044501  042514          .ASCIZ  'FAILED TO READ THAT PATTERN BACK.'
 8315  042574  020104  047524  051040
 8316  042602  040505  020104  044124
 8317  042610  052101  050040  052101
 8318  042616  042524  047122  041040
 8319  042624  041501  027113     000
 8320
 8321  042631     101  020116  047125  EM124:  .ASCII  'AN UNEXPECTED ERROR OCCURRED WHILE RUNNING THE '
 8322  042636  054105  042520  052103
 8323  042644  042105  042440  051122
 8324  042652  051117  047440  041503
 8325  042660  051125  042522  020104
 8326  042666  044127  046111  020105
 8327  042674  052522  047116  047111
 8328  042702  020107  044124  020105
 8329  042710  040515  047111  042524          .ASCII  'MAINTENANCE REGISTER'<CRLF>'COUNT PATTERN '
 8330  042716  040516  041516  020105
 8331  042724  042522  044507  052123
 8332  042732  051105  041600  052517
 8333  042740  052116  050040  052101
 8334  042746  042524  047122     040
 8335  042753     124  051505  027124          .ASCIZ  'TEST. NOTE MISSES WERE BEING FORCED TO BOTH GROUPS.'
 8336  042760  047040  052117  020105
 8337  042766  044515  051523  051505
 8338  042774  053440  051105  020105
 8339  043002  042502  047111  020107
 8340  043010  047506  041522  042105
```

```
8341  043016  052040  020117  047502
8342  043024  044124  043440  047522
8343  043032  050125  027123     000
8344
8345  043037     115  044501  052116  EM127:  .ASCII  'MAINTENANCE REGISTER TEST FAILED.'<CRLF>
8346  043044  047105  047101  042503
8347  043052  051040  043505  051511
8348  043060  042524  020122  042524
8349  043066  052123  043040  044501
8350  043074  042514  027104     200
8351  043101     116  020117  051124          .ASCII  'NO TRAP OR ABORT OCCURRED WHEN THE PATTERN WAS PUT '
8352  043106  050101  047440  020122
8353  043114  041101  051117  020124
8354  043122  041517  052503  051122
8355  043130  042105  053440  042510
8356  043136  020116  044124  020105
8357  043144  040520  052124  051105
8358  043152  020116  040527  020123
8359  043160  052520  020124
8360  043164  047111  052040  042510          .ASCIZ  'IN THE MAINTENANCE REGISTER.'
8361  043172  046440  044501  052116
8362  043200  047105  047101  042503
8363  043206  051040  043505  051511
8364  043214  042524  027122     000
8365
8366  043221     105  051122  051117  EM130:  .ASCIZ  'ERROR REGISTER WILL NOT UNLOCK, OR CLEAR.'
8367  043226  051040  043505  051511
8368  043234  042524  020122  044527
8369  043242  046114  047040  052117
8370  043250  052440  046116  041517
8371  043256  026113  047440  020122
8372  043264  046103  040505  027122
8373  043272     000
8374
8375  043273     105  051122  051117  EM131:  .ASCII  'ERROR REGISTER AND MAINTENANCE REGISTER TEST FAILED.'
8376  043300  051040  043505  051511
8377  043306  042524  020122  047101
8378  043314  020104  040515  047111
8379  043322  042524  040516  041516
8380  043330  020105  042522  044507
8381  043336  052123  051105  052040
8382  043344  051505  020124  040506
8383  043352  046111  042105     056
8384  043357     200  051105  047522          .ASCII  <CRLF>'ERROR REGISTER IS INCORRECTLY SET'
8385  043364  020122  042522  044507
8386  043372  052123  051105  044440
8387  043400  020123  047111  047503
8388  043406  051122  041505  046124
8389  043414  020131  042523     124
8390  043421     200  047506  020122          .ASCIZ  <CRLF>'FOR THE ERROR THAT WAS FORCED USING THE MAINTENANCE REGISTER.'
8391  043426  044124  020105  051105
8392  043434  047522  020122  044124
8393  043442  052101  053440  051501
8394  043450  043040  051117  042503
8395  043456  020104  051525  047111
8396  043464  020107  044124  020105
```

```
 8397  043472  040515  047111  042524
 8398  043500  040516  041516  020105
 8399  043506  042522  044507  052123
 8400  043514  051105  000056
 8401
 8402  043520                          EM140:
 8403  043520  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 8404  043526  046505  051117  020131
 8405  043534  040504  040524  050040
 8406  043542  051101  052111  020131
 8407  043550  044103  041505  042513
 8408  043556  051522  052040  051505
 8409  043564  020124  040506  046111
 8410  043572  042105     056
 8411  043575     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '.
 8412  043602  042514  052040  020117
 8413  043610  047506  041522  020105
 8414  043616  020101  040520  044522
 8415  043624  054524  042440  051122
 8416  043632  051117  020054  051525
 8417  043640  047111  020107
 8418  043644  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 8419  043652  047111  042524  040516
 8420  043660  041516  020105  042522
 8421  043666  044507  052123  051105
 8422  043674  100054
 8423  043676  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY EVEN WORD, LOW BYTE, PARITY '
 8424  043704  046440  044501  020116
 8425  043712  042515  047515  054522
 8426  043720  042440  042526  020116
 8427  043726  047527  042122  020054
 8428  043734  047514  020127  054502
 8429  043742  042524  020054  040520
 8430  043750  044522  054524     040
 8431  043755     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 8432  043762  051105  100054  051040
 8433  043770  040505  044504  043516
 8434  043776  040440  042040  052101
 8435  044004  020101  040520  052124
 8436  044012  051105  020116  044127
 8437  044020  041511  020110
 8438  044024  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 8439  044032  044040  053101  020105
 8440  044040  040503  051525  042105
 8441  044046  040440  020116  051105
 8442  044054  047522  027122     000
 8443
 8444  044061                          EM141:
 8445  044061     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 8446  044066  042515  047515  054522
 8447  044074  042040  052101  020101
 8448  044102  040520  044522  054524
 8449  044110  041440  042510  045503
 8450  044116  051105  020123  042524
 8451  044124  052123  043040  044501
 8452  044132  042514  027104
```

```
 8453  044136  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 8454  044144  020105  047524  043040
 8455  044152  051117  042503  040440
 8456  044160  050040  051101  052111
 8457  044166  020131  051105  047522
 8458  044174  026122  052440  044523
 8459  044202  043516     040
 8460  044205     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 8461  044212  044501  052116  047105
 8462  044220  047101  042503  051040
 8463  044226  043505  051511  042524
 8464  044234  026122     200
 8465  044237     101  020124  044124          .ASCII  'AT THE MAIN MEMORY ODD WORD, LOW BYTE, PARITY '
 8466  044244  020105  040515  047111
 8467  044252  046440  046505  051117
 8468  044260  020131  042117  020104
 8469  044266  047527  042122  020054
 8470  044274  047514  020127  054502
 8471  044302  042524  020054  040520
 8472  044310  044522  054524     040
 8473  044315     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 8474  044322  051105  100054  051040
 8475  044330  040505  044504  043516
 8476  044336  040440  042040  052101
 8477  044344  020101  040520  052124
 8478  044352  051105  020116  044127
 8479  044360  041511  020110
 8480  044364  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 8481  044372  044040  053101  020105
 8482  044400  040503  051525  042105
 8483  044406  040440  020116  051105
 8484  044414  047522  027122     000
 8485
 8486  044421                          EM142:
 8487  044421     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 8488  044426  042515  047515  054522
 8489  044434  042040  052101  020101
 8490  044442  040520  044522  054524
 8491  044450  041440  042510  045503
 8492  044456  051105  020123  042524
 8493  044464  052123  043040  044501
 8494  044472  042514  027104
 8495  044476  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 8496  044504  020105  047524  043040
 8497  044512  051117  042503  040440
 8498  044520  050040  051101  052111
 8499  044526  020131  051105  047522
 8500  044534  026122  052440  044523
 8501  044542  043516     040
 8502  044545     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 8503  044552  044501  052116  047105
 8504  044560  047101  042503  051040
 8505  044566  043505  051511  042524
 8506  044574  026122     200
 8507  044577     101  020124  044124          .ASCII  'AT THE MAIN MEMORY EVEN WORD, HIGH BYTE, PARITY '
 8508  044604  020105  040515  047111
```

```
 8509  044612  046440  046505  051117
 8510  044620  020131  053105  047105
 8511  044626  053440  051117  026104
 8512  044634  044040  043511  020110
 8513  044642  054502  042524  020054
 8514  044650  040520  044522  054524
 8515  044656     040
 8516  044657     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 8517  044664  051105  100054  051040
 8518  044672  040505  044504  043516
 8519  044700  040440  042040  052101
 8520  044706  020101  040520  052124
 8521  044714  051105  020116  044127
 8522  044722  041511  020110
 8523  044726  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 8524  044734  044040  053101  020105
 8525  044742  040503  051525  042105
 8526  044750  040440  020116  051105
 8527  044756  047522  027122     000
 8528
 8529  044763                          EM143:
 8530  044763     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 8531  044770  042515  047515  054522
 8532  044776  042040  052101  020101
 8533  045004  040520  044522  054524
 8534  045012  041440  042510  045503
 8535  045020  051105  020123  042524
 8536  045026  052123  043040  044501
 8537  045034  042514  027104
 8538  045040  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 8539  045046  020105  047524  043040
 8540  045054  051117  042503  040440
 8541  045062  050040  051101  052111
 8542  045070  020131  051105  047522
 8543  045076  026122  052440  044523
 8544  045104  043516     040
 8545  045107     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 8546  045114  044501  052116  047105
 8547  045122  047101  042503  051040
 8548  045130  043505  051511  042524
 8549  045136  026122     200
 8550  045141     101  020124  044124          .ASCII  'AT THE MAIN MEMORY ODD WORD, HIGH BYTE, PARITY '
 8551  045146  020105  040515  047111
 8552  045154  046440  046505  051117
 8553  045162  020131  042117  020104
 8554  045170  047527  042122  020054
 8555  045176  044510  044107  041040
 8556  045204  052131  026105  050040
 8557  045212  051101  052111  020131
 8558  045220  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 8559  045226  026122  020200  042522
 8560  045234  042101  047111  020107
 8561  045242  020101  040504  040524
 8562  045250  050040  052101  042524
 8563  045256  047122  053440  044510
 8564  045264  044103     040
```

```
 8565  045267    123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 8566  045274 020104  040510  042526
 8567  045302 041440  052501  042523
 8568  045310 020104  047101  042440
 8569  045316 051122  051117  000056
 8570
 8571  045324 020040  042524  052123  DH140:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'DATA.'<TAB>'ADDRESS.'
 8572  045332 004456  040503  046114
 8573  045340 040440  020124  041520
 8574  045346 004456  040504  040524
 8575  045354 004456  042101  051104
 8576  045362 051505  027123     000
 8577
 8578         045324                  DH141=DH140
 8579
 8580         045324                  DH142=DH140
 8581
 8582         045324                  DH143=DH140
 8583
 8584  045367    004     003     000  DF140:  .BYTE   4,3,0,2
 8585  045372    002
 8586
 8587         045367                  DF141=DF140
 8588
 8589         045367                  DF142=DF140
 8590
 8591         045367                  DF143=DF140
 8592
 8593         045374                          .EVEN
 8594  045374 001224  001116  001230  DT140:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
 8595  045402 001232  000000
 8596
 8597         045374                  DT141=DT140
 8598
 8599         045374                  DT142=DT140
 8600
 8601         045374                  DT143=DT140
 8602
 8603
 8604  045406 051105  047522  020122  EM132:  .ASCII  'ERROR REGISTER TEST WAS UNABLE TO CAUSE A TIME OUT.'
 8605  045414 042522  044507  052123
 8606  045422 051105  052040  051505
 8607  045430 020124  040527  020123
 8608  045436 047125  041101  042514
 8609  045444 052040  020117  040503
 8610  045452 051525  020105  020101
 8611  045460 044524  042515  047440
 8612  045466 052125     054
 8613  045471    200  052101  040440          .ASCIZ  <CRLF>'AT AN ADDRESS WHICH SHOULD HAVE TIMED OUT.'
 8614  045476 020116  042101  051104
 8615  045504 051505  020123  044127
 8616  045512 041511  020110  044123
 8617  045520 052517  042114  044040
 8618  045526 053101  020105  044524
 8619  045534 042515  020104  052517
 8620  045542 027124     000
```

```
 9621
 9622  045545     105  051122  051117  EM133:  .ASCII  'ERROR REGISTER TEST FAILED.'
 9623  045552  051040  043505  051511
 9624  045560  042524  020122  042524
 9625  045566  052123  043040  044501
 9626  045574  042514  027104
 9627  045600  040600  052106  051105          .ASCII  <CRLF>'AFTER CAUSING A TIME OUT THE ERROR REGISTER SHOULD '
 9628  045606  041440  052501  044523
 9629  045614  043516  040440  052040
 9630  045622  046511  020105  052517
 9631  045630  020124  044124  020105
 9632  045636  051105  047522  020122
 9633  045644  042522  044507  052123
 9634  045652  051105  051440  047510
 9635  045660  046125  020104
 9636  045664  040510  042526  041040          .ASCIZ  'HAVE BEEN SET TO : 000000.'
 9637  045672  042505  020116  042523
 9638  045700  020124  047524  035040
 9639  045706  030040  030060  030060
 9640  045714  027060     000
 9641
 9642  045717     103  047117  051124  EM134:  .ASCII  'CONTROL REGISTER, DISABLE TRAPS, TEST FAILED.'
 9643  045724  046117  051040  043505
 9644  045732  051511  042524  026122
 9645  045740  042040  051511  041101
 9646  045746  042514  052040  040522
 9647  045754  051520  020054  042524
 9648  045762  052123  043040  044501
 9649  045770  042514  027104
 9650  045774  040600  052040  040522          .ASCIZ  <CRLF>'A TRAP OCCURRED WITH BIT 0 SET IN THE CONTROL REGISTER.'
 9651  046002  020120  041517  052503
 9652  046010  051122  042105  053440
 9653  046016  052111  020110  044502
 9654  046024  020124  020060  042523
 9655  046032  020124  047111  052040
 9656  046040  042510  041440  047117
 9657  046046  051124  046117  051040
 9658  046054  043505  051511  042524
 9659  046062  027122     000
 9660
 9661  046065     105  051122  051117  EM135:  .ASCII  'ERROR REGISTER, LOOK UP, TEST FAILED.'
 9662  046072  051040  043505  051511
 9663  046100  042524  026122  046040
 9664  046106  041517  020113  050125
 9665  046114  020054  042524  052123
 9666  046122  043040  044501  042514
 9667  046130  027104
 9668  046132  040600  052106  051105          .ASCII  <CRLF>'AFTER FORCING MULTIPLE ERRORS, TWO, THE ERROR '
 9669  046140  043040  051117  044503
 9670  046146  043516  046440  046125
 9671  046154  044524  046120  020105
 9672  046162  051105  047522  051522
 9673  046170  020054  053524  026117
 9674  046176  052040  042510  042440
 9675  046204  051122  051117     040
 9676  046211     122  043505  051511          .ASCIZ  'REGISTERS WAS INSORRECTLY SET.'
```

```
 8677  046216  042524  051522  053440
 8678  046224  051501  044440  051516
 8679  046232  051117  042522  052103
 8680  046240  054514  051440  052105
 8681  046246  000056
 8682
 8683  046250  052600  042516  050130  EM150:  .ASCIZ  <CRLF>'UNEXPECTED CPU ERROR TRAPPED TO VECTOR ERRVEC (4)!'
 8684  046256  041505  042524  020104
 8685  046264  050103  020125  051105
 8686  046272  047522  020122  051124
 8687  046300  050101  042520  020104
 8688  046306  047524  053040  041505
 8689  046314  047524  020122  051105
 8690  046322  053122  041505  024040
 8691  046330  024464  000041
 8692
 8693                                  ;THESE ARE DATA HEADERS:
 8694
 8695  046334  020040  042524  052123  DH1:    .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'PHYSICAL ADDR.'<TAB>'CALL AT PC.'
 8696  046342  004456  043440  047522
 8697  046350  050125  004456  044120
 8698  046356  051531  041511  046101
 8699  046364  040440  042104  027122
 8700  046372  041411  046101  020114
 8701  046400  052101  050040  027103
 8702  046406     000
 8703  046407     040  052040  051505  DH14:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADDR REG.'
 8704  046414  027124  041411  046101
 8705  046422  020114  052101  050040
 8706  046430  027103  042411  051122
 8707  046436  051117  040440  042104
 8708  046444  020122  042522  027107
 8709  046452  052011  040522  020120          .ASCII  <TAB>'TRAP AT PC.'<TAB>
 8710  046460  052101  050040  027103
 8711  046466     011
 8712  046467     105  051122  051117          .ASCIZ  'ERROR REG.'
 8713  046474  051040  043505  000056
 8714
 8715  046502  020040  042524  052123  DH15:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
 8716  046510  004456  040503  046114
 8717  046516  040440  020124  041520
 8718  046524  000056
 8719
 8720  046526  020040  042524  052123  DH55:   .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'REG ADDRESS.'
 8721  046534  004456  051124  050101
 8722  046542  040440  020124  041520
 8723  046550  004456  040503  046114
 8724  046556  040440  020124  041520
 8725  046564  004456  042522  020107
 8726  046572  042101  051104  051505
 8727  046600  027123     000
 8728
 8729          046526                  DH56=DH55
 8730
 8731          046526                  DH57=DH55
 8732
```

```
 8733            046526                          DH60=DH55
 8734
 8735            046526                          DH61=DH55
 8736
 8737            046526                          DH62=DH55
 8738
 8739  046603     040  052040  051505  DH63:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'CONTROL.'
 8740  046610  027124  041411  046101
 8741  046616  020114  052101  050040
 8742  046624  027103  041411  047117
 8743  046632  051124  046117     056
 8744  046637     115  044501  052116          .ASCIZ  'MAINT.'<TAB>'(DATA READ FROM EACH REGISTER)'
 8745  046644  004456  042050  052101
 8746  046652  020101  042522  042101
 8747  046660  043040  047522  020115
 8748  046666  040505  044103  051040
 8749  046674  043505  051511  042524
 8750  046702  024522     000
 8751
 8752  046705     040  052040  051505  DH64:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'CONTROL REGISTER DATA.'
 8753  046712  027124  041411  046101
 8754  046720  020114  052101  050040
 8755  046726  027103  041411  047117
 8756  046734  051124  046117  051040
 8757  046742  043505  051511  042524
 8758  046750  020122  040504  040524
 8759  046756  000056
 8760
 8761  046760  020040  042524  052123  DH65:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'LOW ORD.'<TAB>'HIGH ORD.'
 8762  046766  004456  040503  046114
 8763  046774  040440  020124  041520
 8764  047002  004456  047514  020127
 8765  047010  051117  027104  044011
 8766  047016  043511  020110  051117
 8767  047024  027104
 8768  047026  024011  040504  040524          .ASCIZ  <TAB>'(DATA READ FROM ADR. REG.)'
 8769  047034  051040  040505  020104
 8770  047042  051106  046517  040440
 8771  047050  051104  020056  042522
 8772  047056  027107  000051
 8773
 8774  047062  020040  042524  052123  DH66:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>'READ.'
 8775  047070  004456  040503  046114
 8776  047076  040440  020124  041520
 8777  047104  004456  051127  052117
 8778  047112  027105  051011  040505
 8779  047120  027104
 8780  047122  042411  050130  041505          .ASCIZ  <TAB>'EXPECTED.'
 8781  047130  042524  027104     000
 8782
 8783  047135     040  052040  051505  DH67:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN READ FROM THE '
 8784  047142  027124  041411  046101
 8785  047150  020114  052101  050040
 8786  047156  027103  050011  052101
 8787  047164  042524  047122  051040
 8788  047172  040505  020104  051106
```

```
 8789  047200  046517  052040  042510
 8790  047206     040
 8791  047207     110  052111  046457  .       .ASCIZ  'HIT/MISS REGISTER.'
 8792  047214  051511  020123  042523
 8793  047222  044507  052123  051105
 8794  047230  000056
 8795
 8796          047135                  DH70=DH67
 8797
 8798          047135                  DH71=DH67
 8799
 8800          047135                  DH72=DH67
 8801
 8802          047135                  DH73=DH67
 8803
 8804          047135                  DH74=DH67
 8805
 8806  047232  020040  042524  052123  DH75:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>
 8807  047240  004456  040503  046114
 8808  047246  040440  020124  041520
 8809  047254  004456  043440  047522
 8810  047262  050125  004456
 8811  047266  042101  051104  051505          .ASCIZ  'ADDRESS.'<TAB>'PATTERN IN CONTROL REG.'
 8812  047274  027123  050011  052101
 8813  047302  042524  047122  044440
 8814  047310  020116  047503  052116
 8815  047316  047522  020114  042522
 8816  047324  027107     000
 8817
 8818          047232                  DH76=DH75
 8819
 8820  047327     040  052040  051505  DH77:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
 8821  047334  027124  041411  046101
 8822  047342  020114  052101  050040
 8823  047350  027103     000
 8824
 8825
 8826          047232                  DH117=DH75
 8827
 8828  047353     040  052040  051505  DH120:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN IN CONTROL REG.'
 8829  047360  027124  041411  046101
 8830  047366  020114  052101  050040
 8831  047374  027103  050011  052101
 8832  047402  042524  047122  044440
 8833  047410  020116  047503  052116
 8834  047416  047522  020114  042522
 8835  047424  027107     000
 8836
 8837  047427     040  052040  051505  DH121:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TEST ADDRESS.'
 8838  047434  027124  041411  046101
 8839  047442  020114  052101  050040
 8840  047450  027103  052011  051505
 8841  047456  020124  042101  051104
 8842  047464  051505  027123     000
 8843
 8844  047471     040  052040  051505  DH122:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>
```

```
 8845  047476  027124  041411  046101
 8846  047504  020114  052101  050040
 8847  047512  027103  053411  047522
 8848  047520  042524  004456
 8849  047524  044124  047105  041440          .ASCIZ  'THEN CLEARED AND READ.'
 8850  047532  042514  051101  042105
 8851  047540  040440  042116  051040
 8852  047546  040505  027104     000
 8853
 8854  047553     040  042524  052123  DH123:  .ASCIZ  ' TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>'READ.'
 8855  047560  004456  040503  046114
 8856  047566  040440  020124  041520
 8857  047574  004456  051127  052117
 8858  047602  027105  051011  040505
 8859  047610  027104     000
 8860
 8861          046407                  DH124=DH14
 8862
 8863  047613     040  052040  051505  DH125:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ADDRESS.'
 8864  047620  027124  041411  046101
 8865  047626  020114  052101  050040
 8866  047634  027103  040411  042104
 8867  047642  042522  051523  000056
 8868
 8869  047650  020040  042524  052123  DH126:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'
 8870  047656  004456  040503  046114
 8871  047664  040440  020124  041520
 8872  047672  004456  051124  050101
 8873  047700  040440  020124  041520
 8874  047706     056
 8875  047707     011  051105  047522          .ASCIZ  <TAB>'ERROR REG.'
 8876  047714  020122  042522  027107
 8877  047722     000
 8878
 8879  047723     040  052040  051505  DH127:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN USED.'
 8880  047730  027124  041411  046101
 8881  047736  020114  052101  050040
 8882  047744  027103  050011  052101
 8883  047752  042524  047122  052440
 8884  047760  042523  027104     000
 8885
 8886  047765     040  052040  051505  DH130:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADR REG.'
 8887  047772  027124  041411  046101
 8888  050000  020114  052101  050040
 8889  050006  027103  042411  051122
 8890  050014  051117  040440  051104
 8891  050022  051040  043505     056
 8892  050027     011  051105  047522          .ASCIZ  <TAB>'ERROR REG.'
 8893  050034  020122  042522  027107
 8894  050042     000
 8895
 8896  050043     040  052040  051505  DH131:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
 8897  050050  027124  041411  046101
 8898  050056  020114  052101  050040
 8899  050064  027103  052011  040522
 8900  050072  020120  052101  050040
```

```
8901  050100  027103     011
8902  050103     105  051122  051117          .ASCIZ  'ERROR ADR REG.'
8903  050110  040440  051104  051040
8904  050116  043505  000056
8905
8906          047613                  DH132=DH125
8907
8908          047650                  DH133=DH126
8909
8910  050122  020040  042524  052123  DH134:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
8911  050130  004456  040503  046114
8912  050136  040440  020124  041520
8913  050144  004456  051124  050101
8914  050152  040440  020124  041520
8915  050160  004456
8916  050162  047503  052116  047522          .ASCIZ  'CONTROL REG.'
8917  050170  020114  042522  027107
8918  050176     000
8919
8920          047327                  DH135=DH77
8921
8922  050177     040  052040  051505  DH150:  .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'CPU ERROR REGISTER.'
8923  050204  027124  052011  040522
8924  050212  020120  052101  050040
8925  050220  027103  041411  046101
8926  050226  020114  052101  050040
8927  050234  027103  041411  052520
8928  050242  042440  051122  051117
8929  050250  051040  043505  051511
8930  050256  042524  027122     000
8931                                  ;THESE ARE DATA FORMAT DESIGNATORS FOR THE DATA TABLE:
8932
8933  050263     004     004     003  DF1:    .BYTE   4,4,3,3
8934  050266     003
8935
8936  050267     004     003     007  DF14:   .BYTE   4,3,7,3,0
8937  050272     003     000
8938
8939  050274     004     003          DF15:   .BYTE   4,3
8940
8941  050276     004     003     003  DF55:   .BYTE   4,3,3,2
8942  050301     002
8943
8944          050276                  DF56=DF55
8945
8946          050276                  DF57=DF55
8947
8948          050276                  DF60=DF55
8949
8950          050276                  DF61=DF55
8951
8952          050276                  DF62=DF55
8953
8954  050302     004     003     000  DF63:   .BYTE   4,3,0,0,0
8955  050305     000     000
8956
```

```
                050302                  DF64=DF63
                050302                  DF65=DF63
                050302                  DF66=DF63
                050302                  DF67=DF63
                050302                  DF70=DF63
                050302                  DF71=DF63
                050302                  DF72=DF63
                050302                  DF73=DF63
                050302                  DF74=DF63
050307   004    003    004     DF75:   .BYTE   4,3,4,2,0
050312   002    000
                050307                  DF76=DF75
050314   004    003    005     DF77:   .BYTE   4,3,5,2,5,0,5,2,5,0
050317   002    005    000
050322   005    002    005
050325   000
                050307                  DF117=DF75
050326   004    003    000     DF120:  .BYTE   4,3,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0
050331   005    000    005
050334   000    005    000
050337   005    000    005
050342   000    005    000
050345   005    000    005
050350   000    005    000
050353   005    000    005
050356   000    005    000
050361   004    003    002     DF121:  .BYTE   4,3,2,2
050364   002
050365   004    003    000     DF122:  .BYTE   4,3,0,0
050370   000
                050365                  DF123=DF122
050371   004    003    007     DF124:  .BYTE   4,3,7,3,0,5,0,
050374   003    000    005
050377   000    000
050401   004    003    002     DF125:  .BYTE   4,3,2,0
050404   000
```

```
9013  050405  004  003  003  DF126:  .BYTE  4,3,3,0,5,2,5,2
9014  050410  000  005  002
9015  050413  005  002
9016
9017  050415  004  003  000  DF127:  .BYTE  4,3,0
9018
9019          050401          DF130=DF125
9020
9021  050420  004  003  003  DF131:  .BYTE  4,3,3,2,5,0,5,0,5,0
9022  050423  002  005  000
9023  050426  005  000  005
9024  050431  000
9025
9026          050401          DF132=DF125
9027
9028          050405          DF133=DF126
9029
9030  050432  004  003  003  DF134:  .BYTE  4,3,3,0,5,2,0
9031  050435  000  005  002
9032  050440  000
9033
9034  050441  004  003  005  DF135:  .BYTE  4,3,5,0,5,0,5,2,5,2
9035  050444  000  005  000
9036  050447  005  002  005
9037  050452  002
9038
9039  050453  004  003  003  DF150:  .BYTE  4,3,3,0
9040  050456  000
9041
9042          050460                  .EVEN
9043
9044                                  ;THESE ARE DATA TABLES:
9045
9046  050460  001224  001226  001230  DT1:    .WORD  $TMP0,$TMP1,$TMP2,$ERRPC,0
9047  050466  001116  000000
9048
9049  050472  001224  001116  001226  DT14:   .WORD  $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,0
9050  050500  001232  001234  000000
9051
9052  050506  001224  001226  000000  DT15:   .WORD  $TMP0,$TMP1,0
9053
9054
9055  050514  001224  001226  001116  DT55:   .WORD  $TMP0,$TMP1,$ERRPC,$TMP3,0
9056  050522  001232  000000
9057
9058          050514          DT56=DT55
9059
9060          050514          DT57=DT55
9061
9062          050514          DT60=DT55
9063
9064          050514          DT61=DT55
9065
9066          050514          DT62=DT55
9067
9068  050526  001224  001116  001230  DT63:   .WORD  $TMP0,$ERRPC,$TMP2,$TMP3,0
```

```
 9069  050534  001232  000000
 9070
 9071  050540  001224  001116  001230  DT64:   .WORD   $TMP0,$ERRPC,$TMP2,0
 9072  050546  000000
 9073
 9074  050550  001224  001116  001230  DT65:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
 9075  050556  001232  000000
 9076
 9077  050562  001224  001116  001230  DT66:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,$TMP4,0
 9078  050570  001232  001234  000000
 9079
 9080          050540                  DT67=DT64
 9081
 9082          050540                  DT70=DT64
 9083
 9084          050540                  DT71=DT64
 9085
 9086          050540                  DT72=DT64
 9087
 9088          050540                  DT73=DT64
 9089
 9090          050540                  DT74=DT64
 9091
 9092  050576  001224  001116  001230  DT75:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP10,$TMP3,0
 9093  050604  001244  001232  000000
 9094
 9095  050612  001224  001116  001230  DT76:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP12,$TMP3,0
 9096  050620  001250  001232  000000
 9097
 9098  050626  001224  001116  033512  DT77:   .WORD   $TMP0,$ERRPC,MTA77,$TMP10,MTB77,$TMP2,MTC77
 9099  050634  001244  033526  001230
 9100  050642  033570
 9101  050644  001250  033625  001232          .WORD   $TMP12,MTD77,$TMP3,0
 9102  050652  000000
 9103
 9104          050612                  DT117=DT76
 9105
 9106  050654  001224  001116  001230  DT120:  .WORD   $TMP0,$ERRPC,$TMP2,MTA120,KCR0,MTG120,KCE0
 9107  050662  033775  006700  034215
 9108  050670  006714
 9109  050672  034025  006702  034215          .WORD   MTB120,KCR1,MTG120,KCE1
 9110  050700  006716
 9111  050702  034055  006704  034215          .WORD   MTC120,KCR2,MTG120,KCE2
 9112  050710  006720
 9113  050712  034105  006706  034215          .WORD   MTD120,KCR3,MTG120,KCE3
 9114  050720  006722
 9115  050722  034135  006710  034215          .WORD   MTE120,KCR4,MTG120,KCE4
 9116  050730  006724
 9117  050732  034165  006712  034215          .WORD   MTF120,KCR5,MTG120,KCE5,0
 9118  050740  006726  000000
 9119
 9120  050744  001224  001116  001230  DT121:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP4,0
 9121  050752  001234  000000
 9122
 9123  050756  001224  001116  001230  DT122:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
 9124  050764  001232  000000
```

```
9125
9126            050756                          DT123=DT122
9127
9128  050770   001224  001116  001226  DT124:  .WORD   $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,MTA124,$TMP6,0
9129  050776   001232  001234  034256
9130  051004   001240  000000
9131
9132  051010   001224  001116  001230  DT125:  .WORD   $TMP0,$ERRPC,$TMP2,0
9133  051016   000000
9134
9135  051020   001224  001116  001230  DT126:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP7,MTA126,$TMP5,MTB126,$TMP3,0
9136  051026   001242  034350  001236
9137  051034   034376  001232  000000
9138
9139            051010                          DT127=DT125
9140
9141  051042   001224  001116  001230  DT130:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP4,0
9142  051050   001234  000000
9143
9144  051054   001224  001116  001230  DT131:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA131,$TMP5
9145  051062   001232  034430  001236
9146  051070   034512  001240  034545          .WORD   MTB131,$TMP6,MTC131,$TMP7,0
9147  051076   001242  000000
9148
9149            051010                          DT132=DT125
9150
9151            051020                          DT133=DT126
9152
9153  051102   001224  001116  001230  DT134:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA134,$TMP4,$TMP6,0
9154  051110   001232  034573  001234
9155  051116   001240  000000
9156
9157  051122   001224  001116  034627  DT135:  .WORD   $TMP0,$ERRPC,MTA135,$TMP2,MTB135,$TMP3
9158  051130   001230  034657  001232
9159  051136   034701  001234  034735          .WORD   MTC135,$TMP4,MTD135,$TMP6,0
9160  051144   001240  000000
9161
9162  051150   001224  001226  001230  DT150:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
9163  051156   001232  000000
9164
9165  051162   000000  000000  000000  BOTTOM: .WORD   0,0,0
9166            057170                          .=.+6000
9167  057170                            BOTPRG:
9168            000001                          .END
```

```
ABORTT  030476          6951    7072#
ADRNG   032767          7118    7544#
BACKAD  030614          7088*   7097    7098#
BIT0    = 000001        1368#   2485    2583    5532    5606    5671    6277    6398
BIT00   = 000001        1359#   1368
BIT01   = 000002        1357#   1367
BIT02   = 000004        1356#   1366
BIT03   = 000010        1355#   1365
BIT04   = 000020        1354#   1364
BIT05   = 000040        1353#   1363
BIT06   = 000100        1352#   1362
BIT07   = 000200        1351#   1361
BIT08   = 000400        1350#   1360    6561
BIT09   = 001000        1349#   1359    6571    6637
BIT1    = 000002        1367#   2490    2588
BIT10   = 002000        1348#   6620
BIT11   = 004000        1347#   6578
BIT12   = 010000        1346#
BIT13   = 020000        1345#   6627
BIT14   = 040000        1344#
BIT15   = 100000        1343#
BIT2    = 000004        1366#   2495    2593
BIT3    = 000010        1365#
BIT4    = 000020        1364#
BIT5    = 000040        1363#
BIT6    = 000100        1362#   2183    7212
BIT7    = 000200        1361#   7192
BIT8    = 000400        1360#
BIT9    = 001000        1359#
BOTPRG  057170          9167#
BOTTOM  051162          2173    7198    9165#
BPTVEC= 000014          1375#
CACHVE= 000114          1382#   2186*   2203*   2295*   2627*   2720*   2811*   3022*   3133*   3201*   3215*   3242*   3318*
                        3403*   3487*   3574*   3655*   3750*   3838*   3937*   4036*   4135*   4234*   4333*   4432*   4531*
                        4634*   4758*   4889*   5001*   5112*   5223*   5334*   5419*   5533*   5607*   5672*   5732*   5756*
                        5870*   5896*   6012*   6037*   6153*   6179*   6270*   6281*   6303*   6391*   6402*   6424*   7052*
                        7058*   7086*
CHAINQ  031034          6516    7197#
CLEAN   030526          6950    7084#
CNRNG   033177          7132    7571#
CONCMS  031742          7196    7424#
CONFLG  030742          2258*   2314*   2321*   2413*   2996*   7128    7154#
CONFL2  030756          2506*   2602*   2665*   2688*   2758*   2781*   2969*   2994    7160#
CONTRL= 177746          1392#   2256    2300*   2302    2395    2438*   2453*   2466*   2497*   2535*   2551*   2564*   2594*
                        2643*   2656*   2667*   2680*   2690*   2736*   2749*   2760*   2773*   2793*   2826*   2828*   2835*
                        2973*   3036*   3054*   3070*   3141*   3214*   3246*   3321*   3406*   3490*   3577*   3659*   3751*
                        3841*   3940*   4039*   4138*   4237*   4336*   4435*   4534*   4667*   4793*   4990*   5002*   5113*
                        5224*   5335*   5531*   5532*   5559    5591*   5606*   5624    5656*   5671*   5689    5729*   5967*
                        6007*   6148*   6272*   6393*   7093*
CPSPUR  030352          2185    4635    5469    7044#   7087
CPUERR= 177766          1405#   2228    2276*   2279*   5467    6645*   7046    7095*
CR      = 000015        1280#   6758    6768
CRLF    = 000200        1281#   2158    6731    6768    7424    7426    7433    7443    7446    7453    7463    7473    7476
                        7486    7492    7495    7500    7507    7512    7519    7529    7532    7539    7544    7559    7571
                        7583    7597    7610    7620    7633    7638    7646    7652    7658    7663    7669    7675    7689
                        7700    7705    7711    7721    7727    7732    7738    7749    7774    7789    7807    7826    7842
```

```
                        7958    7875    7882    7901    7908    7927    7935    7941    7959    7967    8005    8013    8030
                        8038    8061    8069    8101    8118    8127    8151    8160    8176    8196    8214    8238    8267
                        8289    8306    8329    8345    8384    8390    8411    8418    8431    8453    8460    8473    8495
                        8502    8516    8538    8545    8558    8613    8627    8650    8668    8683
DF1       050263        1811    8933#
DF117   = 050307        2049    8986#
DF120     050326        2052    8988#
DF121     050361        2055    8998#
DF122     050365        2058    9001#   9004
DF123   = 050365        2061    9004#
DF124     050371        2064    9006#
DF125     050401        9010#   9019    9026
DF126     050405        9013#   9028
DF127     050415        2073    9017#
DF130   = 050401        2076    9019#
DF131     050420        2080    9021#
DF132   = 050401        2083    9026#
DF133   = 050405        2086    9028#
DF134     050432        2089    9030#
DF135     050441        2092    9034#
DF14      050267        1844    8936#
DF140     045367        2101    8584#   8587    8589    8591
DF141   = 045367        2104    8587#
DF142   = 045367        2107    8589#
DF143   = 045367        2110    8591#
DF15      050274        1947    8939#
DF150     050453        2125    9039#
DF55      050276        1945    8941#   8944    8946    8948    8950    8952
DF56    = 050276        1948    8944#
DF57    = 050276        1951    8946#
DF60    = 050276        1954    8948#
DF61    = 050276        1957    8950#
DF62    = 050276        1960    8952#
DF63      050302        1963    8954#   8957    8959    8961    8963    8965    8967    8969    8971    8973
DF64    = 050302        1966    8957#
DF65    = 050302        1969    8959#
DF66    = 050302        1972    8961#
DF67    = 050302        1975    8963#
DF70    = 050302        1978    8965#
DF71    = 050302        1981    8967#
DF72    = 050302        1984    8969#
DF73    = 050302        1987    8971#
DF74    = 050302        1990    8973#
DF75      050307        1994    8975#   8978    8986
DF76    = 050307        1997    8978#
DF77      050314        2000    8980#
DH1       046334        1811    8695#
DH117   = 047232        2049    8826#
DH120     047353        2052    8828#
DH121     047427        2055    8837#
DH122     047471        2058    8844#
DH123     047553        2061    8854#
DH124   = 046407        2064    8861#
DH125     047613        8863#   8906
DH126     047650        8869#   8908
DH127     047723        2073    8879#
```

```
DH130    047765          2076    8886#
DH131    050043          2080    8896#
DH132 =  047613          2083    8906#
DH133 =  047650          2086    8908#
DH134    050122          2089    8910#
DH135 =  047327          2092    8920#
DH14     046407          1844    8703#   8861
DH140    045324          2101    8571#   8579    8580    8582
DH141 =  045324          2104    8578#
DH142 =  045324          2107    8580#
DH143 =  045324          2110    8582#
DH15     046502          1847    8715#
DH150    050177          2125    8922#
DH55     046526          1945    8720#   8729    8731    8733    8735    8737
DH56  =  046526          1948    8729#
DH57  =  046526          1951    8731#
DH60  =  046526          1954    8733#
DH61  =  046526          1957    8735#
DH62  =  046526          1960    8737#
DH63     046603          1963    8739#
DH64     046705          1966    8752#
DH65     046760          1969    8761#
DH66     047062          1972    8774#
DH67     047135          1975    8783#   8796    8798    8800    8802    8804
DH70  =  047135          1978    8796#
DH71  =  047135          1981    8798#
DH72  =  047135          1984    8800#
DH73  =  047135          1987    8802#
DH74  =  047135          1990    8804#
DH75     047232          1994    8806#   8818    8826
DH76  =  047232          1997    8818#
DH77     047327          2000    8820#   8920
DISPLA=  177570          1275#   6592*   6616*
DT1      050460          1811    9046#
DT117 =  050612          2049    9104#
DT120    050654          2052    9106#
DT121    050744          2055    9120#
DT122    050756          2058    9123#   9126
DT123 =  050756          2061    9126#
DT124    050770          2064    9128#
DT125    051010          9132#   9139    9149
DT126    051020          9135#   9151
DT127 =  051010          2073    9139#
DT130    051042          2076    9141#
DT131    051054          2080    9144#
DT132 =  051010          2083    9149#
DT133 =  051020          2086    9151#
DT134    051102          2089    9153#
DT135    051122          2092    9157#
DT14     050472          1844    9049#
DT140    045374          2101    8594#   8597    8599    8601
DT141 =  045374          2104    8597#
DT142 =  045374          2107    8599#
DT143 =  045374          2110    8601#
DT15     050506          1847    9052#
DT150    051150          2125    9162#
```

```
DT55     050514          1945    9055*   9058    9060    9062    9064    9066
DT56  =  050514          1948    9058*
DT57  =  050514          1951    9060*
DT60  =  050514          1954    9062*
DT61  =  050514          1957    9064*
DT62  =  050514          1960    9066*
DT63     050526          1963    9068*
DT64     050540          1966    9071*   9080    9082    9084    9086    9088    9090
DT65     050550          1969    9074*
DT66     050562          1972    9077*
DT67  =  050540          1975    9080*
DT70  =  050540          1978    9082*
DT71  =  050540          1981    9084*
DT72  =  050540          1984    9096*
DT73  =  050540          1987    9088*
DT74  =  050540          1990    9090*
DT75     050576          1994    9092*
DT76     050612          1997    9095*   9104
DT77     050626          2000    9098*
EMTVEC=  000030          1378#   2139*   2140*
EM1      034763          1811    7763*
EM117    041535          2049    8214*
EM120    041664          2052    8231*
EM121    042077          2055    8257*
EM122    042300          2058    8281*
EM123    042430          2061    8297*
EM124    042631          2064    8321*
EM127    043037          2073    8345*
EM130    043221          2076    8366*
EM131    043273          2080    8375*
EM132    045406          2083    8604*
EM133    045545          2086    8622*
EM134    045717          2089    8642*
EM135    046065          2092    8661*
EM14     035050          1844    7774*
EM140    043520          2101    8402*
EM141    044061          2104    8444*
EM142    044421          2107    8486*
EM143    044763          2110    8529*
EM15     035107          1847    7781*
EM150    046250          2125    8683*
EM55     035157          1945    7788*
EM56     035323          1948    7807*
EM57     035470          1951    7826*
EM60     035612          1954    7842*
EM61     035736          1957    7858*
EM62     036066          1960    7875*
EM63     036214          1963    7892*
EM64     036433          1966    7918*
EM65     036630          1969    7941*
EM66     037213          1972    7986*
EM67     037275          1975    7996*
EM70     037512          1978    8022*
EM71     037770          1981    8053*
EM72     040246          1984    8084*
EM73     040470          1987    8110*
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM74    040754          1990    8143#
EM75    041240          1994    8176#   9194
EM76  = 041240          1997    8194#
EM77    041377          2000    8196#
ERRNG   033077          7125    7558#
ERRVEC= 000004          1371#   2185*   2208    2209*   2222*   2230*   2231    4635*   5446*   6552    6553*   6555*   6558*
                        7087*
ERTYPE  031230          6629    7075    7269#
ERT1    031342          7299#   7363
ERT2    031554          7303    7308    7318    7326    7332    7344    7353    7357#
ERT3    031560          7338    7360#
ERT4    031570          7297    7362    7365#
ERT5    031572          7277    7366#
GNS   = ****** U        1675    2157    6942    6943    6944    6945    6946.   6947    6948    6950    6951    6952    6953
                        6954    6955    6956    6957    6958
HIADRS= 177742          1390#   2244    2354    3206    3270    3279    3286    3355    3364    3372    3439    3448    3456
                        3526    3535    3543    3612    3621    3629    3700    3709    3717    3786    3795    3803    3885
                        3894    3902    3984    3993    4001    4083    4092    4100    4182    4191    4199    4281    4290
                        4298    4380    4389    4397    4479    4488    4496    4578    4587    4595    4703    4721    4729
                        4815    4824    4832    4927    4936    4944    5038    5047    5055    5149    5158    5156    5260
                        5269    5277    5371    5380    5388    5478    5496    5504    5561    5626    5691    5792    5804
                        5813    5932    5944    5953    6073    6085    6094    6215    6227    6236    7061
HIAFLG  030736          2246*   2369*   7114    7152#
HIAFL2  030752          7158#
HIMFLG  030746          2270*   2597*   2999*   7142    7156#
HIMFL2  030762          2500*   2970*   2997    7162#
HITMIS= 177752          1394#   2219    2268    2449    2462    2475    2546    2560    2573    2646    2659    2670    2692
                        2739    2752    2763    2775    2847    3044    3060    3090    3849    3948    4047    4146    4245
                        4344    4443    4542    5597    5662
HMRNG   033407          7146    7597#
HT    = 000011          1278#   6729    6768
IOTVEC= 000020          1376#   2137*   2138*
JA    = 000001          2198#
JADONE  003752          2223    2279#
JAERR   003530          2209    2228#
JAERR0  003540          2230#   2233
JAERR1  003552          2229    2232#
JAERR2  003620          2239    2244#
JAERR3  003640          2245    2250#
JAERR4  003660          2251    2256#
JAERR5  003700          2257    2262#
JAERR6  003720          2263    2268#
JAERR7  003740          2269    2274#
JAERR9  003742          2242    2248    2254    2260    2266    2272    2276#
JATMP   003526          2208*   2222    2225#   2230
JA1     003476          2212    2214#   2220
JA2     003502          2218#   2232    2277
JA3     003514          2222#
JB    = 000002          2290#
JBDONE  004112          2304    2316    2323#
JBERR1  004046          2310#
JBERR2  004076          2308    2318#
JB1     004014          2299    2300#
JB2     004032          2305#
JC    = 000003          2340#
JCDONE  004254          2358    2368    2371#
```

```
JCERR1  004204          2356    2360#
JC1     004156          2350    2352#
JC2     004176          2357#
JD    = 000004          2327    2386#
JDDONE  004272          2407    2415#
JDERR1  004344          2404    2408#
JD1     004316          2397    2398#   2405
JD2     004336          2405#   2414
KA    = 000006          2525#
KADONE  005232          2576    2594#
KAD2    005256          2596    2600#
KAD3    005274          2598    2601    2603#
KAERR1  005162          2548    2580#
KAERR2  005200          2562    2595#
KAERR3  005216          2575    2590#
KAFLG   005160          2534*   2578#   2583*   2588*   2593*   2595    2600
KA1     004760          2535#   2536
KA2     005002          2537    2540#   2541
KA3     005030          2550#   2584
KA4     005054          2552    2554#   2555
KA5     005102          2563#   2589
KA6     005126          2565    2567#   2568
KB    = 000005          2428#   2977
KBDONE  004666          2477    2497#
KBD2    004710          2499    2504#
KBD3    004724          2505    2508#
KBERR1  004616          2450    2482#
KBERR2  004634          2463    2487#
KBERR3  004652          2476    2492#
KBFLG   004614          2437*   2480#   2485*   2490*   2495*   2498    2504
KB1     004426          2438#   2439
KB2     004452          2441    2443#   2444
KB3     004474          2452#   2486
KB4     004520          2454    2456#   2457
KB5     004542          2465#   2491
KB6     004566          2467    2469#   2470
KC    = 000011          2806#
KCCON   006646          2815*   2835    2925*   2928*   2934#   2967
KCDONE  006760          2922    2973#
KCERR   006730          2912    2966#
KCE0    006714          2901    2908    2959#   9106
KCE1    006716          2960#   9109
KCE2    006720          2961#   9111
KCE3    006722          2962#   9113
KCE4    006724          2963#   9115
KCE5    006726          2964#   9117
KCFLG1  006650          2816*   2920*   2936#
KCPTR   006652          2818*   2832    2898    2915*   2916    2938#
KCR0    006700          2890*   2907    2952#   9106
KCR1    006702          2892*   2953#   9109
KCR2    006704          2893*   2954#   9111
KCR3    006706          2894*   2955#   9113
KCR4    006710          2895*   2956#   9115
KCR5    006712          2896*   2957#   9117
KCTBL   006654          2818    2941#
KCTBLB  006676          2916    2950#
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
KC0       006224           2817#   2926    2931
KC1       006240           2817    2823#   2918
KC10      006444           2889#
KC11      006504           2898#
KC12      006522           2902#   2905
KC13      006552           2910#   2913
KC14      006562           2911    2913#
KC15      006564           2915#   2971
KC16      006620           2921    2924#
KC17      006634           2924    2928#
KC2       006322           2837#   2845
KC2.5     006330           2836    2838    2840#
KC3       006362           2833    2848    2862#
KC4       006372           2863    2866#
KC5       006402           2867    2870#
KC6       006412           2871    2874#
KC7       006422           2875    2878#
KC8       006432           2879    2882#
KC9       006442           2883    2886#
KD      = 000007           2622#
KDPAR0= 172360             1548#
KDPAR1= 172362             1549#
KDPAR2= 172364             1550#
KDPAR3= 172366             1551#
KDPAR4= 172370             1552#
KDPAR5= 172372             1553#
KDPAR6= 172374             1554#
KDPAR7= 172376             1555#
KDPDR0= 172320             1526#
KDPDR1= 172322             1527#
KDPDR2= 172324             1528#
KDPDR3= 172326             1529#
KDPDR4= 172330             1530#
KDPDR5= 172332             1531#
KDPDR6= 172334             1532#
KDPDR7= 172336             1533#
KE      = 000010           2715#   2977
KERSTK= 001100             1265#
KF      = 000013           3017#
KFTMP1    007246           3103#
KFTMP2    007250           3026    3104#
KF1       007054           3026#
KF2       007136           3045    3054#
KF3       007174           3061    3070#
KF4       007244           3093    3101#
KF5       007252           3101    3106#
KIPAR0= 172340             1537#   4645    4760    4868    4980    5091    5202    5313    5427    5848    5988    6129    7400
KIPAR1= 172342             1538#
KIPAR2= 172344             1539#
KIPAR3= 172346             1540#
KIPAR4= 172350             1541#
KIPAR5= 172352             1542#
KIPAR6= 172354             1543#   5444#
KIPAR7= 172356             1544#
KIPDR0= 172300             1515#   4647    4762    4870    4982    5093    5204    5315    5429    5850    5990    6131
KIPDR1= 172302             1516#
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
KIPDR2= 172304           1517#
KIPDR3= 172306           1518#
KIPDR4= 172310           1519#
KIPDR5= 172312           1520#
KIPDR6= 172314           1521#
KIPDR7= 172316           1522#
KSP    =%000006          1299#
KTMP1D  005620           2693#
KTMP1E  006150           2786#
KTMP2D  005622           2632     2694#
KTMP2E  006152           2725     2787#
KV     = 000044          5521#
KVDONE  022416           5554     5565#
KVERR   022356           5533     5556#
KV1     022340           5536     5543#
KV2     022346           5546#
KX     = 000045          5581#
KXDONE  022616           5615     5631#
KXERR   022556           5607     5621#
KX1     022520           5598     5606#
KX2     022550           5592     5613#
KY     = 000012          2993#
KY1     007002           2995     2997#
KY2     007016           2998     3000#
KZ     = 000046          5646#
KZDONE  023016           5680     5696#
KZERR   022756           5672     5686#
KZ1     022720           5663     5671#
KZ2     022750           5657     5678#
K1D     005332           2632#
K1E     005662           2725#
K2D     005376           2643#
K2E     005726           2736#
K3D     005436           2647     2654#
K3E     005766           2740     2747#
K4D     005502           2660     2667#
K4E     006032           2753     2760#
K5D     005542           2671     2678#
K5E     006072           2764     2771#
K6D     005612           2677     2683     2690#
K6E     006142           2770     2776     2783#
K7D     005624           2691     2696#
K7E     006154           2784     2789#
LF     = 000012          1279#    6762     6762
LOADRS= 177740           1389#    2211     2238     2353     3205     3269     3276     3285     3354     3362     3371     3438     3446
                         3455     3525     3533     3542     3611     3619     3628     3699     3707     3716     3795     3793     3802
                         3884     3892     3901     3983     3991     4000     4082     4090     4099     4191     4189     4198     4280
                         4288     4297     4379     4387     4396     4478     4486     4495     4577     4585     4594     4702     4720
                         4727     4814     4822     4831     4926     4934     4943     5037     5045     5054     5148     5156     5165
                         5259     5267     5276     5370     5378     5387     5477     5495     5502     5560     5625     5690     5783
                         5791     5803     5811     5923     5931     5943     5951     6064     6072     6084     6092     6206     6214
                         6226     6234     6339     6350     6460     6471     7056     7060
LOAFLG  030734           2204     2240*    2366*    7112     7151#
LOAFL2  030750           7157#
LOC    = 026110          2856#    2857#    2858#    2859     3324#    3325#    3326#    3327     3409#    3410#    3411#    3412     3493#
                         3494#    3495#    3496     3580#    3581#    3582#    3583     3668#    3669#    3670#    3671     3754#    3755#
```

```
                    3756#  3757   4786#  4787#  4788#  4789   4895#  4896#  4897#  4898   5007#  5008#  5009#
                    5010   5118#  5119#  5120#  5121   5229#  5230#  5231#  5232   5340#  5341#  5342#  5343
                    5538#  5539#  5540#  5541   5737#  5738#  5739#  5740   5761#  5762#  5763#  5764   5875#
                    5876#  5877#  5878   5901#  5902#  5903#  5904   6017#  6018#  6019#  6020   6042#  6043#
                    6044#  6045   6158#  6159#  6160#  6161   6184#  6185#  6186#  6187   6327#  6328#  6329#
                    6330   6448#  6449#  6450#  6451
LOOP    003276      2167#  6525
MA    = 000014      3125#
MAB   = 000015      3232#
MABDON  010030      3258   3274   3280   3295#
MABRR0  007640      3242   3260#
MABRR1  007650      3263#
MABRR2  007666      3268#  3277   3279
MABRR3  007722      3266   3276#
MABRR4  007744      3261   3282#
MABR15  007652      3264#  3293
MAB2    007622      3254#
MADONE  007522      3192   3214#
MAERR   007436      3133   3194#
MAERR1  007460      3199   3204#
MAINT = 177750      1393#  2262   2305*  2306   3143   3245   3320   3405   3489   3576   3657   3749   3840
                    3939   4038   4137   4236   4335   4434   4533   4668   4782   4892   5004   5115   5226
                    5337   5535   5610   5675   5734   5758   5872   5898   6014   6039   6155   6181   6283
                    6306   6404   6427   7090*
MANFLG  030744      2264*  2315*  3179*  5914*  6030*  6171*  6197*  7135   7155#
MANFL2  030760      3188*  3210*  3257*  3291*  3340*  3378*  3424*  3462*  3511*  3549*  3597*  3635*  3685*
                    3723*  3771*  3809*  3870*  3908*  3969*  4007*  4069*  4106*  4167*  4205*  4266*  4304*
                    4365*  4403*  4464*  4502*  4563*  4601*  4687*  4708*  4800*  4838*  4912*  4950*  5023*
                    5061*  5134*  5172*  5245*  5283*  5356*  5394*  5750*  5774*  5888*  6055*  7161#
MAPH0 = 170202      1632#
MAPH00= 170202      1568#  1632   5449*
MAPH01= 170206      1570#  1634
MAPH02= 170212      1572#  1636
MAPH03= 170216      1574#  1638
MAPH04= 170222      1576#  1640
MAPH05= 170226      1578#  1642
MAPH06= 170232      1580#  1644
MAPH07= 170236      1582#  1646
MAPH1 = 170206      1634#
MAPH10= 170242      1584#
MAPH11= 170246      1586#
MAPH12= 170252      1588#
MAPH13= 170256      1590#
MAPH14= 170262      1592#
MAPH15= 170266      1594#
MAPH16= 170272      1596#
MAPH17= 170276      1598#
MAPH2 = 170212      1636#
MAPH20= 170302      1600#
MAPH21= 170306      1602#
MAPH22= 170312      1604#
MAPH23= 170316      1606#
MAPH24= 170320      1608#
MAPH25= 170326      1610#
MAPH26= 170332      1612#
MAPH27= 170336      1614#
```

```
MAPH3 = 170216           1638#
MAPH30= 170342           1616#
MAPH31= 170346           1618#
MAPH32= 170352           1620#
MAPH33= 170356           1622#
MAPH34= 170362           1624#
MAPH35= 170366           1626#
MAPH36= 170372           1628#
MAPH37= 170376           1630#
MAPH4 = 170222           1640#
MAPH5 = 170226           1642#
MAPH6 = 170232           1644#
MAPH7 = 170236           1646#
MAPL0 = 170200           1631#
MAPL00= 170200           1567#   1631    4650    4765    4873    4985    5096    5207    5318    5448#   5853    5993    6134
MAPL01= 170204           1569#   1633
MAPL02= 170210           1571#   1635
MAPL03= 170214           1573#   1637
MAPL04= 170220           1575#   1639
MAPL05= 170224           1577#   1641
MAPL06= 170230           1579#   1643
MAPL07= 170234           1581#   1645
MAPL1 = 170204           1633#
MAPL10= 170240           1583#
MAPL11= 170244           1585#
MAPL12= 170250           1587#
MAPL13= 170254           1589#
MAPL14= 170260           1591#
MAPL15= 170264           1593#
MAPL16= 170270           1595#
MAPL17= 170274           1597#
MAPL2 = 170210           1635#
MAPL20= 170300           1599#
MAPL21= 170304           1601#
MAPL22= 170310           1603#
MAPL23= 170314           1605#
MAPL24= 170320           1607#
MAPL25= 170324           1609#
MAPL26= 170330           1611#
MAPL27= 170334           1613#
MAPL3 = 170214           1637#
MAPL30= 170340           1615#
MAPL31= 170344           1617#
MAPL32= 170350           1619#
MAPL33= 170354           1621#
MAPL34= 170360           1623#
MAPL35= 170364           1625#
MAPL36= 170370           1627#
MAPL37= 170374           1629#
MAPL4 = 170220           1639#
MAPL5 = 170224           1641#
MAPL6 = 170230           1643#
MAPL7 = 170234           1645#
MA1     007336           3145    3148#   3191
MA2     007354           3170#
MA3     007402           3173    3181#
```

```
MA4       007426          3182    3190#   3212
MB      = 000016          3308#
MBDONE    010346          3341    3360    3366    3381#
MBERRO    010146          3318    3343#
MB1       010120          3322    3329#
MB2       010124          3331#
MB3       010130          3336#
MC      = 000017          3393#
MCDONE    010662          3425    3444    3450    3465#
MCERRO    010462          3403    3427#
MC1       010434          3407    3414#
MC2       010440          3416#
MC3       010444          3420#
MD      = 000020          3477#
MDDONE    011202          3512    3531    3537    3552#
MDERRO    011002          3487    3514#
MD1       010752          3491    3499#
MD2       010756          3501#
MD3       010764          3507#
ME      = 000021          3564#
MEDONE    011522          3598    3617    3623    3638#
MEERRO    011322          3574    3600#
MEMERR= 177744            1391#   2250    2349*   3198    3204    3260    3264*   3265    3271    3289    3343    3349*   3350
                          3356    3375    3427    3433*   3434    3440    3459    3514    3520*   3521    3527    3546    3600
                          3606*   3607    3613    3632    3688    3694*   3695    3701    3720    3774    3780*   3781    3787
                          3806    3873    3879*   3880    3886    3905    3972    3978*   3979    3985    4004    4071    4077*
                          4078    4084    4103    4170    4176*   4177    4183    4202    4269    4275*   4276    4282    4301
                          4368    4374*   4375    4381    4400    4467    4473*   4474    4480    4499    4566    4572*   4573
                          4579    4598    4695    4706    4715*   4716    4722    4803    4809*   4810    4816    4835    4915
                          4921*   4922    4928    4947    5026    5032*   5033    5039    5058    5137    5143*   5144    5150
                          5169    5248    5254*   5255    5261    5280    5359    5365*   5366    5372    5391    5471    5481
                          5490*   5491    5497    5562    5627    5692    5781    5788    5798*   5799    5805    5921    5928
                          5938*   5939    5945    6062    6069    6079*   6080    6086    6204    6211    6221*   6222    6228
                          6336    6343*   6347    6354*   6457    6464*   6468    6475*   6644*   7053    7059    7094*
ME1       011272          3578    3586#
ME2       011276          3588#
ME3       011304          3593#
MF      = 000022          3650#
MFDONE    012036          3686    3705    3711    3726#
MFERRO    011636          3655    3688#
MF1       011604          3666    3673#
MF2       011616          3659    3679#
MF3       011620          3681#
MG      = 000023          3738#
MGDONE    012356          3772    3791    3797    3812#
MGERRO    012156          3750    3774#
MG1       012124          3752    3759#
MG2       012132          3762#
MG3       012140          3767#
MH      = 000024          3828#
MHDONE    012722          3871    3890    3896    3911#
MHERRO    012522          3838    3873#
MH1       012502          3844    3860#
MH2       012504          3866#
MI      = 000025          3927#
MIDONE    013266          3970    3989    3995    4010#
```

```
MIERRO  013066          3937    3972#
MI1     013046          3943    3959#
MI2     013050          3965#
MJ    = 000026          4026#
MJDONE  013E32          4069    4088    4094    4109#
MJERRO  013432          4036    4071#
MJ1     013412          4042    4058#
MJ2     013414          4064#
MK    = 000027          4125#
MKDONE  014176          4168    4187    4193    4208#
MKERRO  013776          4135    4170#
MK1     013756          4141    4157#
MK2     013760          4163#
ML    = 000030          4224#
MLDONE  014542          4267    4286    4292    4307#
MLERRO  014342          4234    4269#
ML1     014322          4240    4256#
ML2     014324          4262#
MMDES   031132          6952    7229#
MMESRS  031746          7208    7426#
MMRFLG  030740          2252*   3273*   3359*   3443*   3530*   3616*   3704*   3790*   3889*   3988*   4087*   4186*   4285*
                        4384*   4483*   4582*   4724*   4819*   4931*   5042*   5153*   5264*   5375*   5499*   5808*   5948*
                        6089*   6231*   7121    7153#
MMRFL2  030754          3292*   3379*   3463*   3550*   3636*   3724*   3810*   3909*   4008*   4107*   4206*   4305*   4404*
                        4503*   4602*   4839*   4951*   5062*   5173*   5284*   5395*   5483*   7159#
MMR0  = 177572          1416#   1420    3347*   3431*   3518*   3604*   3692*   3778*   3877*   3976*   4075*   4174*   4273*
                        4372*   4471*   4570*   4665*   4713*   4780*   4807*   4888*   4919*   5000*   5030*   5111*   5141*
                        5222*   5252*   5333*   5363*   5443*   5488*   5796*   5895*   5936*   6011*   6034*   6077*   6152*
                        6178*   6219*   7091*   7389
MMR1  = 177574          1417#   1421
MMR2  = 177576          1418#   1422
MMR3  = 172516          1419#   1423    3348*   3432*   3519*   3605*   3693*   3779*   3878*   3977*   4076*   4175*   4274*
                        4373*   4472*   4571*   4664*   4714*   4779*   4808*   4887*   4920*   4999*   5031*   5110*   5142*
                        5221*   5253*   5332*   5364*   5442*   5489*   5797*   5894*   5937*   6010*   6035*   6078*   6151*
                        6177*   6220*   7092*
MMSKIP= 104422          4633    4757    4866    4978    5089    5200    5311    5425    5846    5986    6127    6952#
MMVEC = 000250          1384#
MN    = 000031          4323#
MNRNG   033301          7139    7583#
MO    = 000032          4422#
MODONE  015452          4465    4484    4490    4505#
MOERRO  015252          4432    4467#
MONF    031130          2167*   7202*   7217#
MONTTY  031126          2170*   7209    7215#
MO1     015232          4438    4454#
MO2     015234          4460#
MP    = 000033          4521#
MPDONE  016016          4564    4583    4589    4604#
MPERRO  015616          4531    4566#
MP1     015576          4537    4553#
MP2     015600          4559#
MQ    = 000043          5414#
MQDONE  022242          5465    5500    5507#
MQERR   022036          5446    5467#
MQ1     022016          5461#
MQ2     022052          5468    5471#
```

```
MQ3     022140          5472    5486#
MQ4     022142          5484    5488#
MQ5     022166          5494#   5503    5505
MQ6     022222          5492    5502#
MR    = 000034          4623#
MRDONE  016446          4688    4725    4732#
MRERRO  016240          4634    4690#
MR1     016220          4684#
MR2     016252          4692    4695#
MR3     016342          4696    4711#
MR4     016346          4709    4713#
MR5     016372          4719#   4728    4730
MR6     016426          4717    4727#
MS    = 000035          4747#
MSDONE  017066          4801    4820    4826    4841#
MSERRO  016666          4758    4803#
MSIZER  031154          6953    7245#
MS1     016640          4784    4791#
MS2     016644          4793#
MS3     016650          4796#
MT    = 000036          4856#
MTA101  033670          7633#
MTA11   032136          7453#
MTA120  033775          7646#   9106
MTA124  034256          7688#   9128
MTA126  034350          7700#   9135
MTA131  034430          7711#   9144
MTA134  034573          7732#   9153
MTA135  034627          7738#   9157
MTA17   032203          7461#   7484
MTA20   032237          7470#
MTA21   032246          7473#
MTA43   032333          7486#
MTA45   032406          7495#
MTA5    032054          7443#
MTA50   032464          7507#
MTA77   033512          7610#   9098
MTB120  034025          7652#   9109
MTB126  034376          7705#   9135
MTB131  034512          7721#   9146
MTB135  034657          7744#   9157
MTB17   032210          7463#
MTB21 = 032203          7484#
MTB45   032434          7500#
MTB77   033526          7613#   9098
MTC120  034055          7658#   9111
MTC131  034545          7727#   9146
MTC135  034701          7749#   9159
MTC17   032230          7467#
MTC45   032451          7504#
MTC77   033570          7620#   9098
MTDONE  017512          4913    4932    4938    4953#
MTD120  034105          7663#   9113
MTD135  034735          7755#   9159
MTD77   033625          7626#   9101
MTERRO  017312          4889    4915#
```

```
MTE120  034135          7669#   9115
MTF120  034165          7675#   9117
MTG120  034215          7681#   9106    9109    9111    9113    9115    9117
MT1     017262          4893    4901#
MT2     017274          4908#
MU    = 000037          4968#
MUDONE  020132          5024    5043    5049    5064#
MUERRO  017732          5001    5026#
MU1     017706          5005    5013#
MU2     017714          5019#
MV    = 000040          5079#
MVDONE  020552          5135    5154    5160    5175#
MVERRO  020352          5112    5137#
MV1     020326          5116    5124#
MV2     020334          5130#
MW    = 000041          5190#
MWDONE  021172          5246    5265    5271    5286#
MWERRO  020772          5223    5248#
MW1     020746          5227    5235#
MW2     020754          5241#
MY    = 000042          5301#
MXDONE  021612          5357    5376    5382    5397#
MXERRO  021412          5334    5359#
MX1     021366          5338    5346#
MX2     021374          5352#
MOM1  = 000014          1664#   2438    3141    3246    3321    3406    3490    3577    3658    3751    4667    4830    5531
                        5729    5867    6007    6148    6272    6393
M1MO  = 000014          1663#   1664    4783
NA    = 000047          5719#
NADONE  023402          5751    5775    5809    5816#
NA1     023104          5735    5742#
NA10    023362          5800    5811#
NA2     023110          5744#   5783    5789
NA3     023136          5732    5754#
NA4     023160          5759    5766#
NA5     023164          5768#
NA6     023212          5756    5778#
NA7     023236          5782    5786#
NA8     023302          5784    5796#
NA9     023326          5802#   5812    5814
NB    = 000050          5836#
NBDONE  024072          5889    5915    5949    5956#
NB1     023560          5873    5880#
NB10    024052          5940    5951#
NB2     023564          5882#   5923    5929
NB3     023612          5870    5892#
NB4     023650          5899    5906#
NB5     023654          5908#
NB6     023702          5896    5918#
NB7     023726          5922    5926#
NB8     023772          5924    5936#
NB9     024016          5942#   5952    5954
NC    = 000051          5976#
NCDONE  024572          6031    6056    6090    6097#
NC1     024264          6015    6022#
NC10    024552          6081    6092#
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
NO2       024270          6024*   6064    6070
NO3       024316          6012    6034*
NO4       024350          6040    6047*
NO5       024354          6048*
NO6       024402          6037    6059*
NO7       024426          6063    6067*
NO8       024472          6065    6077*
NO9       024516          6083*   6093    6095
ND      = 000052          6117*
NDDONE    025276          6172    6198    6232    6239*
ND1       024764          6156    6163*
ND10      025256          6223    6234*
ND2       024770          6165*   6206    6212
ND3       025016          6153    6175*
ND4       025054          6182    6189*
ND5       025060          6191*
ND6       025106          6179    6201*
ND7       025132          6205    6209*
ND8       025176          6207    6219*
ND9       025222          6225*   6235    6237
NMDONE    015106          4366    4385    4391    4406*
NMERRO    014706          4333    4368*
NM1       014666          4339    4355*
NM2       014670          4361*
NOCNC     031106          7194    7211*
PARCNT    030764          6276    6397    7172*
PC      =%000007          1302*   6276*   6397*   6503*   6506*   6516*   6520*   6629*   6736*   6743*   6750*   6761*   6766*
                          6993    7028*   7075*   7179*   7206*   7313*   7331*   7343*   7349*   7367*   7413*   7423*
PDMSG1    032512          7512*
PDMSG2    032670          7532*
PIRG    = 177772          1273*
PIROVE  = 000240          1383*
POWERM    032005          6992    7433*
PR0     = 000000          1305*
PR1     = 000040          1306*
PR2     = 000100          1307*
PR3     = 000140          1308*
PR4     = 000200          1309*
PR5     = 000240          1310*
PR6     = 000300          1311*
PR7     = 000340          1312*
PS      = 177776          1270*   1271    2131*
PSW     = 177776          1271*   7096*
PWRVEC  = 000024          1377*   2143*   2144*   6965*   6966*   6974*   6989*   6990*
RESMON    031006          2180    7189*
RESREG  = 104414          6948*   7027    7417
RESVEC  = 000010          1372*
RSET    = 104416          3295    3381    3465    3552    3638    3726    3812    3911    4010    4109    4209    4307    4406
                          4505    4604    4732    4841    4953    5064    5175    5286    5397    5507    5565    6631    6696
                          5816    5956    6097    6239    6361    6482    6950*   7076    7190    7313    7233
R0      =%000000          1284*   1292    2172*   2176*   2205*   2207*   2211*   2215    2219*   2236    2238    2244
                          2250    2256    2262    2268    2302*   2303    2311    2319    2353*   2361    2364    2370
                          2385*   2399*   2400    2406*   2441*   2443    2454*   2456    2467*   2637*   2640*   2644*
                          2554    2565*   2567    2632*   2633*   2634    2638    2725*   2726*   2731    2735*   2737*
                          2833*   2839    2840*   2843    2844*   2866*   2889*   2890    2899*   2902*   2903*   2904    2907*
                          2910    3026*   3030*   3031    3033    3037    3041    3047    3055    3057    3063    3146*   3156
```

```
                    4645*  4657*  4658*  4661*  4760*  4772*  4773*  4776*  4868*  4880*  4881*  4884*  4980*
                    4992*  4993*  4996*  5091*  5103*  5104*  5107*  5202*  5214*  5215*  5218*  5313*  5325*
                    5326*  5329*  5427*  5433*  5434*  5435*  5436*  5437*  5438*  5439*  5440*  5450*  5459
                    5592*  5593   5594   5600   5657*  5658   5659   5665   5848*  5860*  5861*  5864*  5988*
                    6000*  6001*  6004*  6129*  6141*  6142*  6145*  6273*  6288   6298   6309   6319   6356
                    6358*  6394*  6409   6419   6430   6440   6477   6479*  6513*  6517*  6520   6667   6692*
                    6721   6722*  6723   6726*  6862   6872*  6876   6892   6893   6906*  6926   6927*  6928
                    6929*  6930*  6931*  6967   6988*  7017*  7021   7031*  7033*  7035*  7053*  7054   7056*
                    7057   7174   7189*  7192*  7193   7245   7247*  7248*  7249*  7252*  7253*  7255*  7258
                    7259   7262*  7271   7272*  7273*  7279*  7280*  7281*  7282   7288   7295   7298*  7299
                    7304   7310   7320   7327   7333   7340   7346   7360*  7366*  7391*  7392*  7393*  7400*
                    7402
R1      =%000001    1285#  1293   2173*  2175*  2204*  2206*  2306*  2307   2312   2354*  2357   2362   2400*
                    2403   2410   2442*  2444*  2455*  2457*  2469*  2470*  2538*  2541*  2553*  2555*  2566*
                    2568*  2634*  2635*  2636   2644   2645   2669   2727*  2728*  2729   2737   2738   2762
                    2823*  2827   2834*  2845*  2870*  2891*  2892   2898*  2908*  2910   3031*  3032*  3086
                    3088   3095   3143*  3149*  3150   3151*  3156   3172   3331   3416   3501   3588   3679
                    3762   4653*  4654*  4655   4768*  4769*  4770   4793   4876*  4877*  4879   4903   4988*
                    4989*  4990   5099*  5100*  5101   5210*  5211*  5212   5321*  5322*  5323   5428*  5431
                    5546   5744   5768   5856*  5857*  5858   5882   5908   5996*  5997*  5998   6024   6049
                    6137*  6138*  6139   6165   6191   6287*  6288*  6290   6308*  6309*  6312   6408*  6409*
                    6411   6429*  6430*  6433   6668   6691*  6863   6876*  6877   6881   6905*  6968   6987*
                    7013*  7017   7018*  7030*  7032*  7034*  7172*  7174   7177*  7198*  7200   7246   7250*
                    7254*  7261*  7294   7295*  7301   7306   7312   7322   7329   7335   7348   7361   7365*
                    7384*  7385   7397*  7398*  7399*  7406
R10     =%000000    1292#
R11     =%000001    1293#
R12     =%000002    1294#
R13     =%000003    1295#
R14     =%000004    1296#
R15     =%000005    1297#
R2      =%000002    1286#  1294   2174*  2175   2396*  2399   2401   2405*  2409   2449*  2462*  2475*  2483
                    2488   2493   2546*  2547   2560*  2561   2573*  2574   2581   2586   2591   2638*  2639*
                    2640   2657   2658   2681   2731*  2732*  2733   2750   2751   2774   2824*  2829   2832*
                    2837*  2841*  2874*  2893   2900*  2905*  2909*  2913*  3033*  3034*  3086   3088   3097
                    3150*  3181   3186   3245*  3249*  3250*  3320*  3330*  3334*  3405*  3415*  3418*  3489*
                    3500*  3504*  3576*  3587*  3590*  3657*  3674*  3675   3676*  3760*  3764*  3840*  3859*
                    3860*  3939*  3958*  3959*  4038*  4057*  4058*  4137*  4156*  4157*  4236*  4255*  4256*
                    4335*  4354*  4355*  4434*  4453*  4454*  4533*  4552*  4553*  4647*  4652*  4662*  4663*
                    4670*  4672*  4762*  4767*  4777*  4782*  4792*  4794*  4870*  4875*  4885*  4892*  4902*
                    4904*  4982*  4987*  4997*  5004*  5014*  5015*  5093*  5098*  5108*  5115*  5125*  5126*
                    5204*  5209*  5219*  5226*  5236*  5237*  5315*  5320*  5330*  5337*  5347*  5348*  5429*
                    5431*  5535*  5544*  5552*  5610*  5612*  5613*  5675*  5677*  5678*  5734*  5743*  5745*
                    5758*  5767*  5769*  5850*  5855*  5865*  5872*  5881*  5893*  5898*  5907*  5909*  5990*
                    5995*  6005*  6014*  6023*  6025*  6039*  6048*  6050*  6131*  6136*  6146*  6155*  6164*
                    6166*  6181*  6190*  6192*  6277   6283*  6289*  6293*  6294*  6306*  6311*  6314*  6315*
                    6398   6404*  6410*  6414*  6415*  6427*  6432*  6435*  6436*  6669   6690*  6864   6975*
                    6879*  6882   6889*  6890*  6891   6896*  6904*  6969   6986*  7019*  7020   7021*  7036*
                    7037*  7173*  7176*  7199*  7200*  7403*  7404*  7407*  7409
R3      =%000003    1287#  1295   2401*  2402*  2403   2411   2654*  2655*  2656   2663   2678*  2679*  2680
                    2686   2747*  2748*  2749   2756   2771*  2772*  2773   2779   2878*  2894   2901*  2904*
                    4648*  4660*  4763*  4775*  4871*  4883*  4983*  4995*  5094*  5106*  5205*  5217*  5316*
                    5328*  5430*  5432*  5851*  5863*  5991*  6003*  6132*  6144*  6515*  6670   6689*  6803
                    6812*  6818*  6819*  6822*  6827*  6828*  6829   6838*  6865   6873*  6874*  6888*  6891*
                    6900*  6901*  6903*  6970   6985*  7016*  7029*  7036   7191*  7203   7402*  7406*  7410
R4      =%000004    1288#  1296   2882*  2885   3144*  3149   3176   3191   3185   3190*  3244*  3249   3255
```

```
                    3319*   3330    3337    3404*   3415    3421    3488*   3500    3508    3575*   3587    3594    3656*
                    3674    3682    3748*   3749*   3760    3768    3839*   3859    3867    3938*   3958    3966    4037*
                    4057    4065    4136*   4156    4164    4235*   4255    4263    4334*   4354    4362    4433*   4453
                    4461    4532*   4552    4560    4649*   4653    4657    4659*   4764*   4768    4772    4774*   4781*
                    4792    4797    4872*   4876    4880    4882*   4891*   4902    4909    4984*   4988    4992    4994*
                    5003*   5014    5020    5095*   5099    5103    5105*   5114*   5125    5131    5206*   5210    5214
                    5216*   5225*   5236    5242    5317*   5321    5325    5327*   5336*   5347    5353    5534*   5544
                    5609*   5612    5674*   5677    5733*   5743    5757*   5767    5852*   5856    5860    5862*   5871*
                    5881    5897*   5907    5992*   5996    6000    6002*   6013*   6023    6038*   6048    6133*   6137
                    6141    6143*   6154*   6164    6180*   6190    6282*   6289    6305*   6311    6403*   6410    6426*
                    6432    6671    6688*   6804    6806*   6807*   6808*   6809    6810*   6824    6826*   6834*   6937*
                    6971    6984*   7015*   7022*   7197*   7201*
R5      =%000005    1289#   1297    2847*   2866    2870    2874    2878    2882    2886*   2896    3090*   3091*   3092
                    3172*   3177    3659*   3675    3844*   3845    3846    3852    3943*   3944    3945    3951    4042*
                    4043    4044    4050    4141*   4142    4143    4149    4240*   4241    4242    4248    4339*   4340
                    4341    4347    4438*   4439    4440    4446    4537*   4538    4539    4545    4650*   4655*   4656*
                    4765*   4770*   4771*   4873*   4878*   4879*   4985*   4990*   4991*   5096*   5101*   5102*   5207*
                    5212*   5213*   5318*   5323*   5324*   5853*   5958*   5959*   5993*   5998*   5999*   6134*   6139*
                    6140*   6672    6687*   6805    6811*   6813*   6815*   6816*   6817*   6818    6836*   6866    6868*
                    6870*   6877*   6881*   6896    6902*   6972    6983*   7014*   7020*   7025*   7026
R6      =%000006    1290#   1298    2132*   2133*   2134
R7      =%000007    1291#   1302
SAVREG=  104412     6947#   7012    7383
SDPAR0=  172260     1504#
SDPAR1=  172262     1505#
SDPAR2=  172264     1506#
SDPAR3=  172266     1507#
SDPAR4=  172270     1508#
SDPAR5=  172272     1509#
SDPAR6=  172274     1510#
SDPAR7=  172276     1511#
SDPDR0=  172220     1482#
SDPDR1=  172222     1483#
SDPDR2=  172224     1484#
SDPDR3=  172226     1485#
SDPDR4=  172230     1486#
SDPDR5=  172232     1487#
SDPDR6=  172234     1488#
SDPDR7=  172236     1489#
SIPAR0=  172240     1493#
SIPAR1=  172242     1494#
SIPAR2=  172244     1495#
SIPAR3=  172246     1496#
SIPAR4=  172250     1497#
SIPAR5=  172252     1498#
SIPAR6=  172254     1499#
SIPAR7=  172256     1500#
SIPDR0=  172200     1471#
SIPDR1=  172202     1472#
SIPDR2=  172204     1473#
SIPDR3=  172206     1474#
SIPDR4=  172210     1475#
SIPDR5=  172212     1476#
SIPDR6=  172214     1477#
SIPDR7=  172216     1478#
SIZE    =  104424   6953#
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SIZEHI= 177762          1402#
SIZELO= 177760          1400#   7248
SKAD    030524          2200*   2292*   2342*   2388*   2430*   2527*   2624*   2717*   2808*   3019*   3127*   3234*   3310*
                        3395*   3479*   3566*   3652*   3740*   3830*   3929*   4028*   4127*   4226*   4325*   4424*   4523*
                        4625*   4749*   4858*   4970*   5081*   5192*   5303*   5416*   5523*   5583*   5648*   5721*   5838*
                        5978*   6119*   6267*   6388*   7077    7079#   7214    7234
SKBADR  030616          6954    7112#
SKBCNR  030660          6956    7128#
SKBERR  030642          6955    7121#
SKBHMR  030714          6958    7142#
SKBMNR  030676          6957    7135#
SKIPT = 104420          3856    3955    4054    4153    4252    4351    4450    4549    5604    5669    6351#   7049    7065
                        7149
SKPBAD= 104426          2347    6954#
SKPBCN= 104432          2297    2393    2435    2532    2629    2722    2813    3024    3131    3239    3315    3400    3484
                        3571    3745    3835    3934    4033    4132    4231    4330    4429    4528    4630    4754    4863
                        4975    5086    5197    5308    5422    5528    5588    5653    5726    5843    5983    6124    6956#
SKPBER= 104430          2348    3238    3314    3399    3483    3570    3744    3834    3933    4032    4131    4230    4329
                        4428    4527    4629    4753    4862    4974    5085    5196    5307    5421    5527    5587    5652
                        5725    5842    5982    6123    6955#
SKPBHM= 104436          2436    2533    2630    2723    2814    3025    3241    3317    3402    3486    3573    3747    3837
                        3936    4035    4134    4233    4332    4431    4530    4632    4756    4865    4977    5088    5199
                        5310    5424    5530    5590    5655    5728    5845    5985    6126    6958#
SKPBMN= 104434          2298    3132    3240    3316    3401    3485    3572    3746    3836    3935    4034    4133    4232
                        4331    4430    4529    4631    4755    4864    4976    5087    5198    5309    5423    5529    5589
                        5654    5727    5844    5984    6125    6957#
SKRNG   030730          7119    7126    7133    7140    7148#
SP    =%000006          1298#   1299    1300    1301    2136*   2232    2234    2235    3207    3208    3263    3283    3284
                        3346    3369    3370    3430    3453    3454    3517    3540    3541    3603    3626    3627    3691
                        3714    3715    3777    3800    3801    3876    3899    3900    3975    3998    3999    4074    4097
                        4098    4173    4196    4197    4272    4295    4296    4371    4394    4395    4470    4493    4494
                        4569    4592    4593    4643*   4690    4699    4700    4701    4711*   4806    4829    4830    4918
                        4941    4942    5029    5052    5053    5140    5163    5164    5251    5274    5275    5362    5385
                        5386    5475    5476    5486    5557    5558    5622    5623    5687    5698    5780*   5920*   6061*
                        6203*   6342    6353    6463    6474    6510*   6552*   6555    6557    6558    6563*   6564*   6565
                        6588    6589    6593*   6624    6639*   6642*   6667*   6668*   6669*   6670*   6671*   6672*   6673*
                        6674*   6675*   6676*   6683*   6684*   6685*   6686*   6687    6688    6689    6690    6691    6692
                        6721*   6722    6723*   6725    6726    6727*   6729    6731    6733    6737    6739*   6741*   6749*
                        6753    6757    6758    6762    6795*   6796    6797    6798*   6803*   6804*   6805*   6811    6836
                        6837    6838    6839*   6840*   6862*   6863*   6864*   6865*   6866*   6867*   6868    6871*   6884
                        6886*   6888    6898    6900    6902    6903    6904    6905    6906    6908*   6909*   6926*   6927
                        6967*   6968*   6969*   6970*   6971*   6972*   6973    6979*   6983    6984    6985    6986    6987
                        6988    6993*   7013    7026*   7044    7047    7062    7063    7066    7072    7074    7088    7089*
                        7148    7232    7245*   7246*   7247    7259*   7261    7262    7271*   7275*   7294*   7301*   7306*
                        7312*   7314*   7315    7322*   7329*   7348*   7350    7365    7366    7384    7411*   7413*   7414
                        7418*
SPUR    030400          2186    2203    2295    2627    2720    2811    3022    3201    3202    3215    5419    6270    6338
                        6349    6391    6459    6470    7052#   7059    7086
SR0   = 177572          1420#
SR1   = 177574          1421#
SR2   = 177576          1422#
SR3   = 172516          1423#
SSP   =%000006          1300#
STACK = 001100          1264#   1265    1266    1267    2136    7089
START   003010          1679    2130#   6994
STKLMT= 177774          1272#
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
STOP    031072          7205    7207#
SUPSTK= 000700          1266#
SWR     = 177570        1274#   1275    6547*   6561    6563    6571    6578    6617    6620    6627    6631    6637    7229
SW0     = 000001        1340#
SW00    = 000001        1330#   1340
SW01    = 000002        1329#   1339
SW02    = 000004        1328#   1338
SW03    = 000010        1327#   1337
SW04    = 000020        1326#   1336
SW05    = 000040        1325#   1335
SW06    = 000100        1324#   1334
SW07    = 000200        1323#   1333
SW08    = 000400        1322#   1332
SW09    = 001000        1321#   1331
SW1     = 000002        1339#
SW10    = 002000        1320#
SW11    = 004000        1319#
SW12    = 010000        1318#
SW13    = 020000        1317#
SW14    = 040000        1316#
SW15    = 100000        1315#
SW2     = 000004        1338#
SW3     = 000010        1337#
SW4     = 000020        1336#
SW5     = 000040        1335#
SW6     = 000100        1334#
SW7     = 000200        1333#   7229
SW8     = 000400        1332#
SW9     = 001000        1331#
SYSTID= 177764          1404#
SOMOM1= 000034          1662#   2466
SOMI    = 000030        1660#   2564    2654    2736    2744    2771    2825    2928    3054    3841    3940    4237    4336
                        5002    5224    5591    5656
SIMO    = 000044        1659#   2551    2643    2651    2678    2747    2823    2925    3036    4039    4138    4435    4534
                        5113    5335
SIMOM1= 000054          1661#   2453
TAB     = 000011        1658#   7441    7446    7453    7467    7470    7476    7486    7495    7504    7507    7633    7638
                        7700    7705    7711    7721    7727    7732    8571    8695    8703    8709    8715    8720    8739
                        8744    8752    8761    8768    8774    8780    8783    8806    8811    8820    8826    8837    8844
                        8854    8863    8869    8875    8879    8886    8892    8896    8910    8922
TBITVE= 000014          1373#
TESTR1= 140000          1665#   2635    2728    2823    3032
TESTR2= 142000          1666#   2639    2732    2824    3034
TESTR3= 144000          1667#
TKVEC   = 000060        1380#   2170    2180*   2181*   7209*
TOP     003332          2168    2177#
TPVEC   = 000064        1381#
TRAPVE= 000034          1379#   2141*   2142*
TRTVEC= 000014          1374#
TST1    003402          2196#
TST10   005624          2624    2713#
TST11   006154          2717    2804#
TST12   006764          2808    2992#
TST13   007016          3015#
TST14   007262          3019    3123#
TST15   007534          3127    3230#
```

TST16	010032	3234	3306#										
TST17	010350	3310	3391#										
TST2	003756	2200	2289#										
TST20	010664	3395	3475#										
TST21	011204	3479	3562#										
TST22	011524	3566	3648#										
TST23	012040	3652	3736#										
TST24	012360	3740	3826#										
TST25	012724	3830	3925#										
TST26	013270	3929	4024#										
TST27	013634	4028	4123#										
TST3	004112	2292	2338#										
TST30	014200	4127	4222#										
TST31	014544	4226	4321#										
TST32	015110	4325	4420#										
TST33	015454	4424	4519#										
TST34	016020	4523	4621#										
TST35	016450	4625	4745#										
TST36	017070	4749	4854#										
TST37	017514	4858	4966#										
TST4	004254	2342	2383#										
TST40	020134	4970	5077#										
TST41	020554	5081	5188#										
TST42	021174	5192	5299#										
TST43	021614	5303	5412#										
TST44	022244	5416	5519#										
TST45	022420	5523	5579#										
TST46	022620	5583	5644#										
TST47	023020	5648	5717#										
TST5	004372	2388	2426#										
TST50	023404	5721	5834#										
TST51	024074	5838	5974#										
TST52	024574	5978	6115#										
TST53	025300	6119	6263#										
TST54	025654	6267	6384#										
TST55	026230	6388	6484#										
TST6	004724	2430	2523#										
TST7	005274	2527	2620#										
TVADFL	031576	7330*	7342*	7369#	7387								
TVADHI	031602	7376#	7386*	7409*									
TVADLO	031600	7375#	7385*	7410*	7411								
TYPDS =	104410	6511	6946#	7307									
TYPE =	104400	2155	6509	6512	6622	6630	6734	6830	6907	6942#	6991	7117	7124	7131
		7138	7145	7195	7207	7269	7284	7286	7290	7292	7316	7336	7351	7357
		7415												
TYPOC =	104402	6943#	7276	7302										
TYPON =	104406	6945#												
TYPOS =	104404	6944#	7323											
TYPVAD	031604	7331	7343	7383#										
UA =	000054	6265#												
UAER1	025540	6281	6335#											
UAER2	025600	6303	6346#											
UATMP1	025534	6287	6299	6332#	6339									
UATMP2	025536	6308	6320	6333#	6350									
UA1	025342	6275#	6359											
UA2	025366	6278	6281#											

```
UA3     025422          6296#
UA4     025442          6303#   6304    6344
UA5     025506          6317#
UA6     025526          6324#
UA7     025634          6279    6324    6356#
UA8     025652          6357    6361#
UB    = 000055          6386#
UBER1   026114          6402    6456#
UBER2   026154          6424    6467#
UBTMP1  026110          6408    6420    6453#   6460
UBTMP2  026112          6429    6441    6454#   6471
UB1     025716          6396#   6480
UB2     025742          6399    6402#
UB3     025776          6417#
UB4     026016          6424#   6425    6465
UB5     026062          6438#
UB6     026102          6445#
UB7     026210          6400    6445    6477#
UB8     026226          6478    6482#
UDPAR0= 177660          1460#
UDPAR1= 177662          1461#
UDPAR2= 177664          1462#
UDPAR3= 177666          1463#
UDPAR4= 177670          1464#
UDPAR5= 177672          1465#
UDPAR6= 177674          1466#
UDPAR7= 177676          1467#
UDPDR0= 177620          1438#
UDPDR1= 177622          1439#
UDPDR2= 177624          1440#
UDPDR3= 177626          1441#
UDPDR4= 177630          1442#
UDPDR5= 177632          1443#
UDPDR6= 177634          1444#
UDPDR7= 177636          1445#
UIPAR0= 177640          1449#
UIPAR1= 177642          1450#
UIPAR2= 177644          1451#
UIPAR3= 177646          1452#
UIPAR4= 177650          1453#
UIPAR5= 177652          1454#
UIPAR6= 177654          1455#
UIPAR7= 177656          1456#
UIPDR0= 177600          1427#
UIPDR1= 177602          1428#
UIPDR2= 177604          1429#
UIPDR3= 177606          1430#
UIPDR4= 177610          1431#
UIPDR5= 177612          1432#
UIPDR6= 177614          1433#
UIPDR7= 177616          1434#
USESTK= 000600          1267#
USP   =%000006          1301#
$BDADR  001122          1729#
$BDDAT  001126          1731#
$BELL   001300          1785#   6622    6647
```

```
$CHARC  027340          6744*   6751    6760*   6765#
$CMTAG  001100          1717#   2131    2132    2139    2145    2146    2147
$CM1   = 000024         1743#   1744#   1745#   1746#   1747#   1748#   1749#   1750#   1751#   1752#   1753#   1754#   1755#
                        1756#   1757#   1758#   1759#   1760#   1761#   1762#   1763#
$CM2   = 000050         1743#   1744#   1745#   1746#   1747#   1748#   1749#   1750#   1751#   1752#   1753#   1754#   1755#
                        1756#   1757#   1758#   1759#   1760#   1761#   1762#   1763#
$CM3   = 000024         1741#   1743
$CM4   = 000024         1763#   1764#   1765#   1766#   1767#   1768#   1769#   1770#   1771#   1772#   1773#   1774#   1775#
                        1776#   1777#   1778#   1779#   1780#   1781#   1782#   1783#
$CRLF   001305          1787#   6630    6647    6734    6768    7270    7287    7293
$DBLK   030006          6873    6907    6915#
$DB22O  030232          7012#   7313    7349    7412
$DOAGN  026344          6505    6514    6518    6524#
$DTBL   027776          6876    6911#
$ENDAD  026334          1703    2153    6520#   6634
$ENDCT  026264          2145    6507#
$ENDMG  026350          6509    6526#
$ENULL  026365          6512    6529#
$EOP    026230          6497#
$EOPCT  026256          2145*   6504#   6508
$ERFLG  001103          1720#   6537    6567    6569    6575*   6596    6614*   6647
$ERMAX  001115          1726#   2148*   6569    6591*   6596
$ERROR  026644          2139    6613#
$ERRPC  001116          1727#   6624*   6625*   6626    6647    7275    8594    9046    9049    9055    9068    9071    9074
                        9077    9092    9095    9098    9106    9120    9123    9128    9132    9135    9141    9144    9153
                        9157
$ERRTB  001310          1805#   7281
$ERTTL  001112          1724#   6623*   6647
$ESCAP  001276          1784#   2147*   6590*   6640    6642    6647
$FILLC  001150          1739#   6737    6768
$FILLS  001147          1738#   6768
$GDADR  001120          1728#
$GDDAT  001124          1730#
$GET42  026306          6513#
$HD    = 000003         1241    1242
$ICNT   001104          1721#   6582*   6583    6585*   6595
$ILLUP  030224          6965    6996#
$ITEMB  001114          1725#   6626*   6647    7073*   7273
$LF     001306          1788#   6647    6768
$LPADR  001106          1722#   2149*   6573*   6588*   6593    6595
$LPERR  001110          1723#   2150*   2212*   2299*   2350*   2397*   2439*   2452*   2465*   2536*   2550*   2563*   2817*
                        3145*   6275*   6304*   6396*   6425*   6573    6589*   6595    6639
$MXCNT  026642          6586    6595#
$NULL   001146          1737#   6739    6768
$NWTST= 000001          2189#   2191    2281#   2293    2325#   2327    2374#   2376    2418#   2420    2510#   2512    2606#
                        2608    2699#   2701    2792#   2794    2975#   2977    3002#   3004    3108#   3110    3220#   3222
                        3298#   3300    3383#   3385    3467#   3469    3554#   3556    3640#   3642    3728#   3730    3815#
                        3817    3914#   3916    4013#   4015    4112#   4114    4211#   4213    4310#   4312    4409#   4411
                        4508#   4510    4610#   4612    4734#   4736    4843#   4845    4955#   4957    5066#   5068    5177#
                        5179    5288#   5290    5399#   5401    5509#   5511    5569#   5571    5634#   5636    5702#   5704
                        5819#   5821    5959#   5961    6100#   6102    6242#   6244    6363#   6365
$OCNT   027566          6802*   6831*   6844#
$OCTVL  030334          7014    7039#
$OMODE  027570          6797*   6801*   6806    6809*   6820*   6946#
$OVER   026626          6548    6566    6574    6584    6592#
$PASS   001100          1718#   6501*   6502*   6510    6526    6580    6596
```

```
$PWRAD  030220          6994#
$PWRDN  030076          2143    6965#   6989
$PWRMG  030214          6992#
$PWRUP  030144          6974    6979#
$QUES   001204          1786#   6647
$RDCHR= ****** U        6947
$RDDEC= ****** U        6947
$RDLIN= ****** U        6947
$RDOCT= ****** U        6947
$REGAD  001152          1741#
$REG0   001154          1743#
$REG1   001156          1744#
$REG10  001174          1751#
$REG11  001176          1752#
$REG12  001200          1753#
$REG13  001202          1754#
$REG14  001204          1755#
$REG15  001206          1756#
$REG16  001210          1757#
$REG17  001212          1758#
$REG2   001160          1745#
$REG20  001214          1759#
$REG21  001216          1760#
$REG22  001220          1761#
$REG23  001222          1762#
$REG3   001162          1746#
$REG4   001164          1747#
$REG5   001166          1748#
$REG6   001170          1749#
$REG7   001172          1750#
$RESRE  027072          6682#   6948
$SAVRE  027034          6666#   6947
$SAVR6  030230          6973*   6979    6980*   6981*   6998#
$SCOPE  026370          2137    6546#
$SETUP= 000037          1652#   2137    2139    2141    2143    2145    2146    2147    2149    2153    6499    6631
$STUP = 177777          1652#
$SVLAD  026600          6556    6587#
$SVPC = 000204          1701#   1706
$SWR  = 167400          1241    1242#   1243#   1251    1252    1253    1254    1255    1256    1257    1783    1784    1795
                        2146    2147    2149    2150    2197    2290    2339    2384    2427    2524    2621    2714    2805
                        2993    3016    3124    3231    3307    3392    3476    3563    3649    3737    3827    3926    4025
                        4124    4223    4322    4421    4520    4622    4746    4855    4967    5078    5189    5300    5413
                        5520    5580    5645    5718    5835    5975    6116    6264    6385    6494    6500    6515    6525
                        6526    6538    6539    6540    6541    6542    6547    6559    6561    6562    6567    6568    6569
                        6576    6577    6578    6589    6592    6595    6604    6605    6606    6607    6608    6609    6617
                        6620    6627    6631    6637    6647
$SWRMK= 000200          1244#   1257    1258    6542    6543    6563    6564
$TAB    032052          7358    7441#
$TIMES  001274          1783#   2146*   2197*   2339*   2384*   2427*   2524*   2621*   2714*   2805*   3016*   3124*   3231*
                        3307*   3392*   3476*   3563*   3649*   3737*   3827*   3926*   4025*   4124*   4223*   4322*   4421*
                        4520*   4622*   4746*   4855*   4967*   5078*   5189*   5300*   5413*   5520*   5580*   5645*   5718*
                        5835*   5975*   6116*   6264*   6385*   6500*   6576*   6583    6586*   6595
$TKB    001140          1734#   2182*   7189    7211*
$TKS    001136          1733#   2134    2183*   7212*
$TMPO   001224          1763#   2202*   2294*   2344*   2390*   2432*   2529*   2626*   2719*   2810*   3021*   3129*   3236*
                        3312*   3397*   3481*   3568*   3654*   3742*   3832*   3931*   4030*   4129*   4228*   4327*   4426*
```

```
                    4525*  4627*  4751*  4860*  4972*  5083*  5194*  5305*  5418*  5525*  5585*  5650*  5723*
                    5840*  5980*  6121*  6269*  6390*  8594   9046   9049   9052   9055   9068   9071   9074
                    9077   9092   9095   9098   9106   9120   9123   9128   9132   9135   9141   9144   9153
                    9157   9162
$TMP1   001226      1764#  2234*  2360*  3048*  3064*  3205*  3853*  3952*  4051*  4150*  4249*  4348*  4447*
                    4546*  5601*  5666*  7044*  7060*  7072*  9046   9049   9052   9055   9128   9162
$TMP10  001244      1771#  2636*  2729*  9092   9098
$TMP11  001246      1772#  2637*  2730*
$TMP12  001250      1773#  2640*  2733*  9095   9101
$TMP13  001252      1774#  2641*  2734*
$TMP14  001254      1775#
$TMP15  001256      1776#
$TMP16  001260      1777#
$TMP17  001262      1778#
$TMP2   001230      1765#  2311*  2319*  2361*  2409*  2483*  2488*  2493*  2581*  2586*  2591*  2650*  2674*
                    2685*  2743*  2767*  2778*  2967*  3047*  3063*  3095*  3176*  3185*  3206*  3255*  3269*
                    3283*  3337*  3354*  3369*  3421*  3438*  3453*  3508*  3525*  3540*  3594*  3611*  3626*
                    3682*  3699*  3714*  3768*  3785*  3800*  3852*  3867*  3884*  3899*  3951*  3966*  3983*
                    3998*  4050*  4065*  4082*  4097*  4149*  4164*  4181*  4196*  4248*  4263*  4280*  4295*
                    4347*  4362*  4379*  4394*  4446*  4461*  4478*  4493*  4545*  4560*  4577*  4592*  4685*
                    4700*  4720*  4797*  4814*  4829*  4909*  4926*  4941*  5020*  5037*  5052*  5131*  5148*
                    5163*  5242*  5259*  5274*  5353*  5370*  5385*  5462*  5475*  5495*  5557*  5600*  5622*
                    5665*  5687*  5748*  5772*  5787*  5803*  5886*  5912*  5927*  5943*  6028*  6053*  6068*
                    6084*  6169*  6195*  6210*  6226*  6298*  6319*  6419*  6440*  7045*  7061*  8594   9046
                    9068   9071   9074   9077   9092   9095   9098   9106   9120   9123   9132   9135   9141
                    9144   9153   9157   9162
$TMP20  001264      1779#
$TMP21  001266      1780#
$TMP22  001270      1781#
$TMP23  001272      1782#
$TMP3   001232      1766#  2236*  2312*  2362*  2410*  2651*  2663*  2675*  2686*  2744*  2756*  2768*  2779*
                    3096*  3177*  3186*  3207*  3270*  3285*  3355*  3371*  3439*  3455*  3526*  3542*  3612*
                    3628*  3700*  3716*  3786*  3802*  3885*  3901*  3984*  4000*  4083*  4099*  4182*  4198*
                    4281*  4297*  4380*  4396*  4479*  4495*  4578*  4594*  4702*  4721*  4815*  4831*  4927*
                    4943*  5038*  5054*  5149*  5165*  5260*  5276*  5371*  5387*  5463*  5477*  5496*  5559*
                    5624*  5689*  5788*  5804*  5928*  5944*  6069*  6085*  6211*  6227*  6299*  6320*  6420*
                    6441*  7046*  7062*  8594   9049   9055   9068   9074   9077   9092   9095   9101   9123
                    9128   9135   9144   9153   9157   9162
$TMP4   001234      1767#  2237*  2411*  3097*  3204*  3271*  3286*  3356*  3372*  3440*  3456*  3527*  3543*
                    3613*  3629*  3701*  3717*  3787*  3803*  3886*  3902*  3985*  4001*  4084*  4100*  4183*
                    4199*  4282*  4298*  4381*  4397*  4480*  4496*  4579*  4595*  4703*  4722*  4816*  4832*
                    4928*  4944*  5039*  5055*  5150*  5166*  5261*  5277*  5372*  5388*  5478*  5497*  5560*
                    5625*  5690*  5789*  5805*  5929*  5945*  6070*  6086*  6212*  6228*  6300*  6321*  6421*
                    6442*  7059*  9049   9077   9120   9128   9141   9153   9159
$TMP5   001236      1768#  3098*  3287*  3373*  3457*  3544*  3630*  3718*  3804*  3903*  4002*  4101*  4200*
                    4299*  4398*  4497*  4596*  4704*  4833*  4945*  5056*  5167*  5278*  5389*  5479*  5561*
                    5626*  5691*  5790*  5930*  6071*  6213*  9135   9144
$TMP6   001240      1769#  3288*  3374*  3458*  3545*  3631*  3719*  3805*  3904*  4003*  4102*  4201*  4300*
                    4399*  4498*  4597*  4705*  4834*  4946*  5057*  5168*  5279*  5390*  5480*  5562*  5627*
                    5692*  5791*  5931*  6072*  6214*  9128   9146   9153   9159
$TMP7   001242      1770#  3289*  3375*  3459*  3546*  3632*  3720*  3806*  3905*  4004*  4103*  4202*  4301*
                    4400*  4499*  4598*  4706*  4835*  4947*  5058*  5169*  5280*  5391*  5481*  5792*  5932*
                    6073*  6215*  9135   9146
$TN   = 000055      1241#  2189   2197#  2198   2199   2281   2290#  2291   2325   2339#  2340   2341   2374
                    2384#  2386   2387   2418   2427#  2428   2429   2510   2524#  2525   2526   2606   2621#
                    2622   2623   2699   2714#  2715   2716   2792   2805#  2806   2807   2975   2993#  3002
```

```
                    3016#  3017   3018   3108   3124#  3125   3126   3220   3231#  3232   3233   3298   3307#
                    3308   3309   3383   3392#  3393   3394   3467   3476#  3477   3478   3554   3563#  3564
                    3565   3640   3649#  3650   3651   3728   3737#  3738   3739   3815   3827#  3828   3829
                    3914   3926#  3927   3928   4013   4025#  4026   4027   4112   4124#  4125   4126   4211
                    4223#  4224   4225   4310   4322#  4323   4324   4409   4421#  4422   4423   4508   4520#
                    4521   4522   4610   4622#  4623   4624   4734   4746#  4747   4748   4843   4855#  4856
                    4857   4955   4967#  4968   4969   5066   5078#  5079   5080   5177   5189#  5190   5191
                    5288   5300#  5301   5302   5399   5413#  5414   5415   5509   5520#  5521   5522   5569
                    5580#  5581   5582   5634   5645#  5646   5647   5702   5718#  5719   5720   5819   5835#
                    5836   5837   5959   5975#  5976   5977   6100   6116#  6117   6118   6242   6264#  6265
                    6266   6363   6385#  6386   6387
$TPB     001144     1736#  6757*  6768
$TPFLG   001151     1740#  6717   6768
$TPS     001142     1735#  6755   6768
$TRAP    030016     2141   6926#
$TRP   = 000040     6933#  6943#  6944#  6945#  6946#  6947#  6948#  6949#  6950   6951#  6952#  6953#  6954#
                    6955#  6956#  6957#  6958#  6959#
$TRPAD   030036     6930   6941#
$TSTNM   001102     1719#  2130*  2202   2294   2344   2390   2432   2529   2626   2719   2810   3021   3129
                    3236   3312   3397   3481   3568   3654   3742   3832   3931   4030   4129   4238   4327
                    4426   4525   4627   4751   4860   4972   5083   5194   5305   5418   5525   5585   5650
                    5723   5840   5980   6121   6269   6390   6499*  6537   6565   6587*  6592   6596   6616
                    6647
$TYPBN= ****** U    6947
$TYPDS   027572     6861#  6946
$TYPE    027130     6717#  6933   6942
$TYPEC   027274     6736   6743   6750   6755#  6756
$TYPEX   027342     6761   6763   6766#
$TYPOC   027370     6800#  6943
$TYPON   027404     6799   6802#  6945
$TYPOS   027344     6795#  6944
$XTSTR   026376     6550#
$$GET4= 000001      6515#  6517#
$$TRP  = 000002     6932#  6943   6944   6945   6946   6947   6948   6949   6951   6952   6953   6954   6955
                    6956   6957   6958   6959
$OFILL   027567     6796*  6800*  6810   6845#
.      = 057170     1671#  1675   1677#  1701   1702#  1704#  1706#  1715#  1789   2135   2149   2150   2856
                    2859#  3161   3324   3327#  3409   3412#  3493   3496#  3580   3583#  3668   3671#  3754
                    3757#  4786   4789#  4895   4898#  5007   5010#  5118   5121#  5229   5232#  5340   5343#
                    5538   5541#  5737   5740#  5761   5764#  5875   5878#  5901   5904#  6017   6020#  6042
                    6045#  6158   6161#  6184   6187#  6327   6330#  6448   6451#  6526   6530   6595   6596
                    6647   6768   6915#  6976   6997   7039#  8593#  9042#  9166#
```

B01

```
MAINDEC-11 DEKBC-B      KMP 11/70 CACHE DIAGNOSTIC PART 1       MACY11 27(732)  09-SEP-76  17:25  PAGE 211
DEKBCB.P11       CROSS REFERENCE TABLE -- MACRO NAMES

ADDCPA   1219#
ADJLST   1205#   2855    3323    3408    3492    3579    3667    3753    4785    4894    5006    5117    5228    5339    5537
         5736    5760    5874    5900    6016    6041    6157    6183    6326    6447
BYTLT1   1208#
BYTLT2   1207#
CHAIN    1188#   6615
CLRMAC   1210#   7084
CLRRF    1204#   2204
CMACPA   1216#
CNVCPA   1217#
CNVMAP   1212#
COMMEN      1#   1650#
ENDCOM      1#   1650#
ERCLR    1189#   6644
ERROR    1959#   2241    2247    2253    2259    2265    2271    2313    2320    2363    2412    2484    2489    2494    2582
         2587    2592    2652    2664    2676    2687    2745    2757    2769    2780    2968    3049    3065    3100    3178
         3187    3209    3256    3272    3290    3339    3358    3377    3423    3442    3461    3510    3529    3548    3596
         3615    3634    3684    3703    3722    3770    3789    3808    3854    3869    3888    3907    3953    3968    3987
         4005    [illegible] 4067 4086   4105    4151    4166    4195    4204    4250    4265    4294    4303    4349    4364
         [illegible] 4429 4448 4463     4482    4501    4547    4562    4581    4600    4686    4693    4707    4723    4799
         4818    4837    4911    4930    4949    5022    5041    5060    5133    5152    5171    5244    5263    5282    5355
         5374    5393    5464    5482    5498    5563    5602    5629    5667    5694    5749    5773    5794    5807    5887
         5913    5934    5947    6029    6054    6075    6088    6170    6196    6217    6230    6301    6322    6422    6443
         7048    7064
ESCAPE      1#   1650#
ITMAC1   1194#
ITEMAC   1206#   1810
JJM1     1225#
KKM1     1215#
KKM2     1216#
KMAC1    1206#   2606    2699
KMAC5    1199#   5582    5647
MMAC1    1194#   3342    3426    3513    3599    3687    3773    3872    3971    4070    4169    4268    4367    4466    4565
         4802    4914    5025    5136    5247    5358
MMAC2    1195#   3335    3419    3506    3592    3680    3766    3865    3964    4063    4162    4261    4360    4459    4558
         4795    4907    5018    5129    5240    5351
MMAC3    1196#   3829    3928    4027    4126    4225    4324    4423    4522
MMAC4    1197#   4969    5080    5191    5302
MMAC5    1198#   4857
MMM1     1222#
MMM2     1223#
MSG      2188#   2191    2281#   2283    2373#   2376    2417#   2420    2510#   2512    2606#   2608    2699#   2701    2792#
         2794    3002#   3004    3108#   3110    3219#   3222    3298#   3300    3383#   3385    3467#   3469    3554#   3556
         3640#   3642    3728#   3730    3815#   3817    3914#   3916    4013#   4015    4112#   4114    4211#   4213    4310#
         4312    4409#   4411    4508#   4510    4610#   4612    4734#   4736    4843#   4845    4955#   4957    5066#   5068
         5177#   5179    5288#   5290    5399#   5401    5509#   5511    5569#   5571    5634#   5636    5702#   5704    5819#
         5821    5959#   5961    6100#   6102    6242#   6244    6363#   6365
MSG101   5700#   5712    5714    5829    5971
MSG102   5701#   5831    5969    6110    6112
MSG20    2605#   2608    2701
MSG200   6242#   6244    6365
MSG25    3217#   3300    3385    3469    3556    3642    3730
MSG27    3814#   3817    3916    4015    4114    4213    4312    4411    4510
MSG30    2975#   2977
MSG300   8402#   8444    8486    8529
MSG31    4607#   4612    4736    4845    4957    5068    5179    5290
```

C01


```
MSG32     4608#   4736    4945
MSG33     4609#   4957    5068    5179    5290
MSG6      2325#   2327
MSG98     5567#   5571    5636
MSG99     5699#   5704    5821    5961    6102
MULT         1#   1650#
NEWTST       1#   1650#   2189    2281    2325    2374    2418    2510    2606    2699    2792    2975    3002    3108    3220
          3298    3383    3467    3554    3640    3728    3815    3914    4013    4112    4211    4310    4409    4508    4610
          4734    4843    4955    5066    5177    5288    5399    5509    5569    5634    5702    5819    5959    6100    6242
          6363
NMAC1     1200#   5731    5754    5869    6036
NMAC2     1201#   5892    6009    6150    6175
NMAC3     1203#   5778    5918    6059    6201
POP          1#   1650#   6687    6902    6983
PUSH         1#   1650#   6667    6861    6967
QGM1      1224#
SCOPE     1269#   2196    2289    2338    2383    2426    2523    2620    2713    2804    2992    3015    3123    3230    3306
          3391    3475    3562    3648    3736    3826    3925    4024    4123    4222    4321    4420    4519    4621    4745
          4854    4966    5077    5188    5299    5412    5519    5579    5644    5717    5834    5974    6115    6263    6384
          6498
SETMAP    1211#   4644    4759    4867    4979    5090    5201    5312    5847    5987    6128
SETMM     1214#   5426
SETTRA    6933#   6943    6944    6945    6946    6947    6948    6950    6951    6952    6953    6954    6955    6956    6957
          6958
SETUP        1#   1650#   2131
SKBREX    1204#   2296    2346    2392    2434    2531    2628    2721    2812    3023    3130    3237    3313    3398    3482
          3569    3743    3833    3932    4031    4130    4229    4328    4427    4526    4628    4752    4861    4973    5084
          5195    5306    5420    5526    5586    5651    5724    5841    5981    6122    6271    6392
SKIP         1#   1650#
SLASH        1#   1650#
SPACE     1650#
SSKAD     1219#   2199    2291    2341    2387    2429    2526    2623    2716    2807    3018    3126    3233    3309    3394
          3478    3565    3651    3739    3829    3928    4027    4126    4225    4324    4423    4522    4624    4748    4857
          4969    5080    5191    5302    5415    5522    5582    5647    5720    5837    5977    6118    6256    6387
STARS        1#   1650#   1681    1708    1789    2189    2195    2281    2288    2325    2337    2374    2382    2418    2425
          2510    2522    2606    2619    2699    2712    2792    2803    2975    2991    3002    3014    3108    3122    3220
          3229    3298    3305    3383    3390    3467    3474    3554    3561    3640    3647    3728    3735    3815    3825
          3914    3924    4013    4023    4112    4122    4211    4221    4310    4320    4409    4419    4508    4518    4610
          4620    4734    4744    4843    4853    4955    4965    5066    5076    5177    5187    5288    5298    5399    5411
          5509    5518    5569    5578    5634    5643    5702    5716    5819    5833    5959    5973    6100    6114    6242
          6262    6363    6383    6487    6531    6597    6648    6695    6769    6848    6917    6960    7000
THIT      1220#   3042    3058    3847    3946    4045    4144    4243    4342    4441    4540    5595    5660
TOMAPO    1209#
TOPAR3    1213#
TRMTRP    6933#
TSTHIT    1221#   3040    3056
TYPBIN       1#   1650#
TYPDEC       1#   1650#   6510
TYPNAM       1#   1650#   2151
TYPNUM       1#   1650#
TYPOCS       1#   1650#
TYPOCT       1#   1650#
TYPTXT       1#   1650#
UMAC1     1189#   6242    6363
UMAC2     1190#   6335    6346    6456    6467
UMAC3     1192#   6296    6317    6417    6438
```

```
$$CMRE   1708*  1743  1744  1745  1746  1747  1748  1749  1750  1751  1752  1753  1754  1755  1756
         1757   1758  1759  1760  1761  1762
$$CMTM   1709*  1763  1764  1765  1766  1767  1768  1769  1770  1771  1772  1773  1774  1775  1776
         1777   1778  1779  1780  1781  1782
$$ESCA      1*  1650*
$$NEWT      1*  1650*  2189  2281  2325  2374  2418  2510  2606  2699  2792  2575  3002  3108  3220
         3298   3383  3467  3554  3640  3728  3815  3914  4013  4112  4211  4310  4409  4508  4610
         4734   4843  4955  5066  5177  5288  5399  5509  5569  5634  5702  5819  5959  6100  6242
         6363
$$SET    6933*  6943  6944  6945  6946  6947  6948  6950  6951  6952  6953  6954  6955  6956  6957
         6958
$$SKIP      1*  1650*
.EQUAT      1*
.HEADE      1*  1228*  1231
.KT11       1*
.SETUP      1*  1228*  1652
.SWRHI      1*  1228*  1246
.SWRLO   1258*
.$ACTI      1*  1228*  1681
.$CATC      1*  1228*  1668
.$CMTA      1*  1228*  1708
.$DB2D      1*
.$DB2O      1*  1230*  7000
.$DIV       1*
.$EOP       1*  1229*  6487
.$ERRO      1*  1229*  6597
.$ERRT      1*
.$MULT      1*
.$POWE      1*  1230*  6960
.$RAND      1*
.$RDDE      1*
.$RDOC      1*
.$READ      1*
.$SAVE      1*  1229*  6648
.$SB2D      1*
.$SB2O      1*
.$SCOP      1*  1229*  6531
.$SIZE      1*
.$SUPR      1*
.$TRAP      1*  1230*  6917
.$TYPB      1*
.$TYPD      1*  1230*  6848
.$TYPE      1*  1229*  6695
.$TYPO      1*  1229*  6769
.1170       1*  1228*  1260
```

```
ADC     7407
ADD     2218  2635  2639  2728  2732  2840  2844  2915  3032  3034  3190  4658  4659  4711  4773
        4774  4881  4882  4993  4994  5104  5105  5215  5216  5326  5327  5780  5861  5862  5920
        6001  6002  6061  6142  6143  6203  6358  6479  6727  6798  6808  6881  7037  7281  7314
        7400  7406  7413
ASH     4654  4769  4877  4989  5100  5211  5322  5857  5997  6138  7280
ASHC    2902  7392  7404
ASL     2837  2841  7177  7252  7393
ASLB    6886
ASR     7029
BCC     2838  2842  6887  7178
BEQ     2154  2304  2308  2358  2365  2368  2505  2601  2683  2776  2911  2917  2924  2995  2998
        3093  3161  3173  3182  3266  3351  3435  3522  3608  3696  3782  3881  3980  4079  4178
        4277  4376  4475  4574  4692  4696  4717  4811  4923  5034  5145  5256  5367  5472  5492
        5784  5800  5924  5940  6065  6081  6207  6223  6337  6348  6357  6458  6469  6478  6514
        6518  6562  6566  6568  6572  6581  6615  6621  6638  6641  6730  6763  6825  7024  7055
        7175  7283  7289  7362  7388  7390
BGE     6584
BGT     6505  6832  6895  7023
BHI     6570
BIC     2402  2633  2655  2679  2726  2748  2772  2889  2891  2903  3030  6502  6564  6822  7036
        7192
BIS     2485  2490  2495  2583  2588  2593  5532  5606  5671  6827  6828  6889  6890  7255
BISB    2183  7212
BIT     2228  2600  2646  2659  2670  2682  2739  2752  2763  2775  3044  3060  3092  3156  3199
        3849  3948  4047  4146  4245  4344  4443  4542  5467  5597  5662  6277  6398  6561  6571
        6578  6620  6627  6637  7054  7174  7229  7389
BITB    6751
BLOS    2220
BLT     6742  6833  6878  6894
BMI     6548  6885
BNE     2135  2152  2168  2229  2233  2239  2245  2251  2257  2263  2269  2356  2404  2450  2463
        2476  2499  2548  2562  2575  2596  2647  2660  2671  2740  2753  2764  3045  3061  3191
        3199  3261  3277  3279  3344  3363  3365  3428  3447  3449  3515  3534  3536  3601  3620
        3622  3689  3708  3710  3775  3794  3796  3850  3874  3893  3895  3949  3973  3992  3994
        4048  4072  4091  4093  4147  4171  4190  4192  4246  4270  4289  4291  4345  4369  4388
        4390  4444  4468  4487  4489  4543  4567  4586  4588  4728  4730  4804  4823  4825  4916
        4935  4937  5027  5046  5048  5138  5157  5159  5249  5268  5270  5360  5379  5381  5468
        5503  5505  5598  5663  5782  5812  5814  5922  5952  5954  6063  6093  6095  6205  6235
        6237  6278  6340  6351  6399  6461  6472  6579  6628  6635  6724  6732  6738  6752  6759
        6823  6883  6982  7113  7115  7122  7129  7136  7143  7194  7205  7230  7274  7296  7300
        7305  7311  7321  7328  7334  7341  7347
BPL     2921  6618  6632  6718  6756  6821  6869  6899
BR      2156  2242  2248  2254  2260  2266  2272  2316  2407  2414  2486  2491  2584  2589  2598
        2677  2691  2770  2784  2836  2848  2863  2867  2871  2875  2879  2883  3101  3192  3212
        3258  3274  3280  3293  3322  3341  3360  3366  3380  3407  3425  3444  3450  3464  3491
        3512  3531  3537  3551  3578  3598  3617  3623  3637  3666  3686  3705  3711  3725  3752
        3772  3791  3797  3811  3871  3890  3896  3910  3970  3989  3995  4009  4069  4088  4094
        4108  4168  4187  4193  4207  4267  4286  4292  4306  4366  4385  4391  4405  4465  4484
        4490  4504  4564  4583  4589  4603  4688  4709  4725  4784  4801  4820  4826  4840  4893
        4913  4932  4938  4952  5005  5024  5043  5049  5063  5116  5135  5154  5160  5174  5227
        5246  5265  5271  5285  5338  5357  5376  5382  5396  5465  5484  5500  5536  5554  5615
        5680  5735  5751  5759  5775  5809  5873  5889  5899  5915  5949  6015  6031  6040  6056
        6090  6156  6172  6182  6198  6232  6324  6445  6550  6556  6559  6574  6577  6720  6735
        6745  6754  6761  6799  6814  6835  6880  6897  6976  6997  7039  7119  7126  7133  7140
        7303  7308  7318  7326  7332  7338  7344  7353  7363
```

```
CLC     7396
CLR     2130  2133  2146  2147  2182  2206  2276  2279  2300  2305  2396  2406  2437  2497  2534
        2535  2594  2637  2641  2667  2690  2730  2734  2760  2783  2815  2899  2973  3070  3096
        3098  3144  3151  3214  3250  3334  3347  3348  3418  3431  3432  3504  3518  3519  3530
        3604  3605  3676  3692  3693  3764  3778  3779  3860  3877  3878  3959  3976  3977  4058
        4075  4076  4157  4174  4175  4256  4273  4274  4355  4372  4373  4454  4471  4472  4553
        4570  4571  4649  4656  4672  4713  4714  4764  4771  4794  4807  4808  4872  4879  4904
        4919  4920  4984  4991  5015  5030  5031  5095  5102  5126  5141  5142  5206  5213  5237
        5252  5253  5317  5324  5348  5363  5364  5433  5488  5489  5552  5613  5678  5745  5769
        5790  5796  5797  5852  5859  5883  5909  5930  5936  5937  5992  5999  6025  6034  6035
        6050  6071  6077  6078  6133  6140  6166  6192  6213  6219  6220  6273  6293  6294  6300
        6314  6315  6321  6394  6414  6415  6421  6435  6436  6442  6499  6500  6576  6590  6645
        6812  6872  6875  6980  7019  7090  7091  7092  7093  7095  7096  7173  7191  7211  7249
        7272  7342  7386  7391  7403
CLRB    6575  6760  6901
CMP     2134  2153  2219  2232  2238  2244  2250  2256  2262  2268  2355  2357  2364  2367  2403
        2498  2547  2561  2574  2595  2910  2916  3086  3088  3181  3260  3263  3276  3278  3343
        3346  3362  3364  3427  3430  3446  3448  3514  3517  3533  3535  3600  3603  3619  3621
        3675  3688  3691  3707  3709  3774  3777  3793  3795  3873  3876  3892  3894  3972  3975
        3991  3993  4071  4074  4090  4092  4170  4173  4189  4191  4269  4272  4288  4290  4368
        4371  4387  4389  4467  4470  4486  4488  4566  4569  4585  4587  4690  4695  4699  4701
        4727  4729  4803  4806  4822  4824  4915  4918  4934  4936  5026  5029  5045  5047  5137
        5140  5156  5158  5248  5251  5267  5269  5359  5362  5378  5380  5471  5486  5502  5504
        5781  5782  5811  5813  5921  5923  5951  5953  6062  6064  6092  6094  6204  6206  6234
        6236  6290  6312  6336  6339  6342  6347  6350  6353  6356  6411  6433  6457  6460  6463
        6468  6471  6474  6477  6557  6583  6634  6893  7047  7063  7066  7074  7148  7193  7203
        7232  7258
CMPB    6565  6569  6729  6731  6737  6758  6762  7304  7310  7320  7327  7333  7340  7346
COM     3091
DEC     2920  6503  7022  7279
DECB    6741  6744  6820  6831
EMT     1268
HALT    1675  2274  2861  2864  2865  2868  2869  2872  2873  2876  2877  2880  2881  2884  2895
        3162  6619  6633  6636  6719  6975  6996  7210  7355
INC     2151  2167  6501  6582  6623  6826  6834  6879  6981  7025  7176  7360
INCB    6587  6614  6764
IOT     1269
JMP     1679  2223  2231  2277  2477  2576  2912  2918  2922  2926  2931  2971  3202  5469  6279
        6338  6344  6349  6359  6400  6459  6465  6470  6480  6525  7067  7077  7097  7214  7234
        7277  7297
JSR     6276  6397  6516  6520  6629  6736  6743  6750  7075  7313  7331  7343  7349  7412
MOV     2131  2132  2136  2137  2138  2139  2140  2141  2142  2143  2144  2145  2149  2150  2170
        2172  2173  2174  2175  2177  2180  2181  2185  2186  2197  2200  2203  2204  2205  2209
        2209  2211  2212  2222  2230  2234  2236  2237  2240  2246  2252  2258  2264  2270  2292
        2295  2299  2302  2306  2311  2312  2314  2315  2319  2321  2339  2342  2349  2350  2353
        2354  2360  2361  2362  2366  2369  2384  2388  2395  2397  2399  2400  2401  2409  2410
        2411  2413  2427  2430  2438  2439  2441  2442  2449  2452  2453  2454  2455  2462  2465
        2466  2467  2468  2475  2483  2488  2493  2500  2506  2524  2527  2536  2537  2538  2546
        2550  2551  2552  2553  2560  2563  2564  2565  2566  2573  2581  2586  2591  2597  2602
        2621  2624  2627  2632  2634  2636  2638  2640  2643  2650  2651  2654  2656  2663  2665
        2674  2675  2678  2680  2685  2686  2688  2714  2717  2720  2725  2727  2729  2731  2733
        2736  2743  2744  2747  2749  2756  2758  2767  2768  2771  2773  2778  2779  2781  2805
        2808  2811  2816  2817  2818  2823  2824  2825  2826  2828  2832  2833  2834  2835  2847
        2866  2870  2874  2878  2882  2886  2890  2892  2893  2894  2895  2896  2898  2900  2901
        2904  2907  2908  2909  2925  2928  2967  2969  2970  2996  2999  3016  3019  3022  3026
        3031  3033  3036  3047  3048  3054  3063  3064  3090  3095  3097  3124  3127  3133  3141
```

DEKBCB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
3143  3145  3146  3149  3150  3172  3176  3177  3179  3185  3186  3188  3201  3204  3205
3206  3207  3210  3215  3231  3234  3242  3244  3245  3246  3249  3255  3257  3264  3269
3270  3271  3273  3283  3285  3286  3287  3288  3289  3291  3292  3307  3310  3318  3319
3320  3321  3330  3337  3340  3349  3354  3355  3356  3359  3369  3371  3372  3373  3374
3375  3378  3379  3392  3395  3403  3404  3405  3406  3415  3421  3424  3433  3438  3439
3440  3443  3453  3455  3456  3457  3458  3459  3462  3463  3476  3479  3487  3488  3489
3490  3500  3508  3511  3520  3525  3526  3527  3530  3540  3542  3543  3544  3545  3546
3549  3550  3563  3566  3574  3575  3576  3577  3587  3594  3597  3606  3611  3612  3613
3616  3626  3628  3629  3630  3631  3632  3635  3636  3649  3652  3655  3656  3657  3658
3659  3674  3682  3685  3694  3699  3700  3701  3704  3714  3716  3717  3718  3719  3720
3723  3724  3737  3740  3748  3749  3750  3751  3760  3768  3771  3780  3785  3786  3787
3790  3800  3802  3803  3804  3805  3806  3809  3810  3827  3830  3838  3839  3840  3841
3844  3852  3853  3859  3867  3870  3879  3884  3885  3886  3889  3899  3901  3902  3903
3904  3905  3908  3909  3926  3929  3937  3938  3939  3940  3943  3951  3952  3958  3966
3969  3978  3983  3984  3985  3988  3998  4000  4001  4002  4003  4004  4007  4008  4025
4028  4036  4037  4038  4039  4042  4050  4051  4057  4065  4068  4077  4082  4083  4084
4087  4097  4099  4100  4101  4102  4103  4106  4107  4124  4127  4135  4136  4137  4138
4141  4149  4150  4156  4164  4167  4176  4181  4182  4183  4186  4196  4198  4199  4200
4201  4202  4205  4206  4223  4226  4234  4235  4236  4237  4240  4248  4249  4255  4263
4266  4275  4280  4281  4282  4285  4295  4297  4298  4299  4300  4301  4304  4305  4322
4325  4333  4334  4335  4336  4339  4347  4348  4354  4362  4365  4374  4379  4380  4381
4384  4394  4396  4397  4398  4399  4400  4403  4404  4421  4424  4432  4433  4434  4435
4438  4446  4447  4453  4461  4464  4473  4478  4479  4480  4483  4493  4495  4496  4497
4498  4499  4502  4503  4520  4523  4531  4532  4533  4534  4537  4545  4546  4552  4560
4563  4572  4577  4578  4579  4582  4592  4594  4595  4596  4597  4598  4601  4602  4622
4625  4634  4635  4643  4645  4647  4648  4650  4652  4653  4655  4657  4661  4662  4664
4665  4667  4668  4670  4685  4687  4700  4702  4703  4704  4705  4706  4708  4715  4720
4721  4722  4724  4746  4749  4758  4760  4762  4763  4765  4767  4768  4770  4772  4776
4777  4779  4780  4781  4782  4783  4792  4797  4800  4809  4814  4815  4816  4819  4829
4831  4832  4833  4834  4835  4838  4839  4855  4858  4868  4870  4871  4873  4875  4876
4878  4880  4884  4885  4887  4888  4889  4890  4891  4892  4902  4909  4912  4921  4926
4927  4928  4931  4941  4943  4944  4945  4946  4947  4950  4951  4967  4970  4980  4982
4983  4985  4987  4988  4990  4992  4996  4997  4999  5000  5001  5002  5003  5004  5014
5020  5023  5032  5037  5038  5039  5042  5052  5054  5055  5056  5057  5058  5061  5062
5078  5081  5091  5093  5094  5096  5098  5099  5101  5103  5107  5108  5110  5111  5112
5113  5114  5115  5125  5131  5134  5143  5148  5149  5150  5153  5163  5165  5166  5167
5168  5169  5172  5173  5189  5192  5202  5204  5205  5207  5209  5210  5212  5214  5218
5219  5221  5222  5223  5224  5225  5226  5236  5242  5245  5254  5259  5260  5261  5264
5274  5276  5277  5278  5279  5280  5283  5284  5300  5303  5313  5315  5316  5318  5320
5321  5323  5325  5329  5330  5332  5333  5334  5335  5336  5337  5347  5353  5356  5365
5370  5371  5372  5375  5385  5387  5388  5389  5390  5391  5394  5395  5413  5416  5419
5427  5428  5429  5430  5431  5434  5435  5436  5437  5438  5439  5440  5442  5443  5444
5446  5448  5449  5450  5462  5463  5475  5477  5478  5479  5480  5481  5483  5490  5495
5496  5497  5499  5520  5523  5531  5533  5534  5535  5544  5557  5559  5560  5561  5562
5580  5583  5591  5592  5600  5601  5607  5609  5610  5612  5622  5624  5625  5626  5627
5645  5648  5656  5657  5665  5666  5672  5674  5675  5677  5687  5689  5690  5691  5692
5718  5721  5729  5732  5733  5734  5743  5748  5750  5756  5757  5758  5767  5772  5774
5787  5788  5789  5791  5792  5798  5803  5804  5805  5808  5835  5838  5848  5850  5851
5853  5855  5856  5858  5860  5864  5865  5867  5870  5871  5872  5881  5886  5888  5894
5895  5896  5897  5898  5907  5912  5914  5927  5928  5929  5931  5932  5938  5943  5944
5945  5948  5975  5978  5988  5990  5991  5993  5995  5996  5998  6000  6004  6005  6007
6010  6011  6012  6013  6014  6023  6028  6030  6037  6038  6039  6048  6053  6055  6068
6069  6070  6072  6073  6079  6084  6085  6086  6089  6116  6119  6129  6131  6132  6134
6136  6137  6139  6141  6145  6146  6148  6151  6152  6153  6154  6155  6164  6169  6171
6177  6178  6179  6180  6181  6190  6195  6197  6210  6211  6212  6214  6215  6221  6226
6227  6228  6231  6264  6267  6270  6272  6275  6281  6282  6283  6287  6288  6289  6298
```

```
            6299  6303  6304  6305  6306  6308  6309  6311  6319  6320  6343  6354  6385  6388  6391
            6393  6396  6402  6403  6404  6408  6409  6410  6419  6420  6424  6425  6426  6427  6429
            6430  6432  6440  6441  6464  6475  6506  6510  6513  6515  6517  6552  6553  6555  6558
            6563  6573  6585  6586  6588  6589  6592  6593  6616  6624  6639  6642  6644  6667  6668
            6669  6670  6671  6672  6673  6674  6675  6676  6683  6684  6685  6686  6687  6688  6689
            6690  6691  6692  6721  6722  6726  6739  6795  6803  6804  6805  6811  6818  6836  6837
            6838  6839  6840  6862  6863  6864  6865  6866  6867  6868  6873  6876  6896  6902  6903
            6904  6905  6906  6908  6909  6926  6927  6930  6965  6966  6967  6968  6969  6970  6971
            6972  6973  6974  6979  6983  6984  6985  6986  6987  6988  6989  6990  6993  7013  7014
            7015  7016  7017  7018  7021  7026  7044  7045  7046  7052  7053  7056  7058  7059  7060
            7061  7062  7072  7086  7087  7088  7089  7094  7172  7189  7197  7198  7199  7200  7202
            7209  7245  7246  7247  7248  7250  7259  7261  7262  7271  7275  7282  7288  7294  7295
            7298  7301  7306  7312  7315  7322  7329  7330  7335  7348  7350  7365  7366  7384  7385
            7402  7409  7410  7411  7414  7418
MOVB        2148  2202  2294  2344  2390  2432  2529  2626  2719  2810  3021  3129  3236  3312  3397
            3481  3568  3654  3742  3832  3931  4030  4129  4228  4327  4426  4525  4627  4751  4860
            4972  5083  5194  5305  5418  5525  5585  5650  5723  5840  5980  6121  6269  6390  6591
            6626  6723  6749  6757  6796  6797  6800  6801  6802  6806  6809  6810  6829  6871  6874
            6888  6891  6900  6929  7020  7073  7273
NEG         6807  6870
NOP         2214  2301  2352  2398  2445  2446  2447  2448  2458  2459  2460  2461  2471  2472  2473
            2474  2542  2543  2544  2545  2556  2557  2558  2559  2569  2570  2571  2572  2668  2761
            2862  3085  3148  3170  3248  3329  3414  3498  3499  3505  3585  3586  3591  3673  3679
            3759  3761  3765  3858  3957  4056  4155  4254  4353  4452  4551  4669  4791  4900  4901
            4905  5012  5013  5123  5124  5234  5235  5345  5346  5458  5543  5545  5553  5611  5614
            5676  5679  5742  5746  5766  5770  5880  5884  5906  5910  6022  6026  6047  6051  6163
            6167  6189  6193  6310  6431  6521  6522  6523
RESET       6519
ROL         6547  6813  6815  6816  6817  6819  7253
ROR         7030  7031  7032  7033  7034  7035  7397  7398  7399
RTI         6594  6646  6677  6693  6728  6841  6910  6995  7116  7123  7130  7137  7144  7231  7263
RTS         6766  6931  7028  7179  7206  7367  7420
SOB         2176  2207  2405  2444  2457  2470  2541  2555  2568  2830  2845  2905  2913  4660  4775
            4883  4995  5106  5217  5328  5432  5863  6003  6144  7201  7254
SUB         6625  6877
TRAP        6933  6943  6944  6945  6946  6947  6948  6950  6951  6952  6953  6954  6955  6956  6957
            6958
TST         2215  2235  2303  2307  2443  2456  2469  2504  2540  2554  2567  2644  2645  2657  2658
            2669  2681  2737  2738  2750  2751  2762  2774  2827  2829  2839  2843  2994  2997  3037
            3041  3055  3057  3208  3265  3284  3331  3350  3370  3416  3434  3454  3501  3521  3541
            3588  3607  3627  3678  3695  3715  3762  3781  3801  3845  3846  3980  3900  3944  3945
            3979  3999  4043  4044  4078  4098  4142  4143  4177  4197  4241  4242  4276  4296  4340
            4341  4375  4395  4439  4440  4474  4494  4538  4539  4573  4593  4716  4793  4810  4830
            4903  4922  4942  5033  5053  5144  5164  5255  5275  5366  5386  5459  5476  5491  5546
            5558  5593  5594  5623  5658  5659  5688  5744  5768  5799  5882  5908  5939  6024  6049
            6080  6165  6191  6222  6554  6580  6617  6631  6640  6725  6733  6753  6824  6882  6892
            6928  7057  7112  7114  7121  7128  7135  7142  7361  7387
TSTB        6567  6717  6755  6884  6898  7299
.ASCII      1786  1787  7443  7453  7473  7486  7512  7519  7522  7532  7544  7558  7571  7583  7597
            7633  7688  7788  7795  7807  7814  7826  7842  7858  7875  7892  7901  7918  7927  7941
            7950  7959  7967  7973  7996  8005  8022  8030  8038  8053  8061  8069  8084  8092  8110
            8118  8127  8143  8151  8160  8176  8196  8214  8231  8238  8257  8267  8281  8289  8297
            8306  8321  8329  8345  8351  8375  8384  8403  8411  8418  8423  8431  8445  8453  8460
            8465  8473  8487  8495  8502  8507  8516  8530  8538  8545  8550  8558  8604  8622  8627
            8642  8661  8668  8703  8709  8739  8761  8774  8783  8806  8844  8869  8886  8896  8910
.ASCIZ      1785  1788  2158  6526  7424  7426  7433  7441  7446  7461  7463  7467  7470  7476  7492
```

```
            7495  7500  7504  7507  7529  7539  7554  7567  7580  7593  7606  7610  7613  7620  7626
            7638  7646  7652  7658  7663  7669  7675  7681  7696  7700  7705  7711  7721  7727  7732
            7738  7744  7749  7755  7763  7774  7781  7804  7823  7833  7849  7865  7882  7908  7935
            7982  7986  8013  8045  8076  8101  8135  8168  8184  8203  8220  8246  8276  8314  8335
            8360  8366  8390  8438  8480  8523  8565  8571  8613  8636  8650  8676  8683  8695  8712
            8715  8720  8744  8752  8768  8780  8791  8811  8820  8828  8837  8849  8854  8863  8875
            8879  8892  8902  8916  8922
.BLKB        7039
.BLKW        6915
.BYTE        1719  1720  1725  1726  1737  1738  1739  1740  6529  6842  6843  6844  6845  7324  7325
            8564  8933  8936  8939  8941  8954  8975  8980  8988  8998  9001  9006  9010  9013  9017
            9021  9030  9034  9039
.ENABL          1  1227
.END         9168
.ENDC        1236  1254  1256  1257  1258  1260  1649  1652  1680  1682  1704  1706  1709  1716  1741
            1763  1783  1784  1785  1786  1789  1790  1813  1816  1819  1822  1825  1828  1831  1834
            1837  1840  1843  1846  1849  1852  1855  1858  1861  1864  1867  1870  1873  1876  1879
            1882  1886  1889  1892  1895  1898  1901  1904  1907  1910  1913  1916  1919  1922  1925
            1928  1931  1934  1937  1940  1943  1947  1950  1953  1956  1959  1962  1965  1968  1971
            1974  1977  1980  1983  1986  1989  1992  1996  1999  2002  2005  2008  2011  2014  2017
            2020  2023  2026  2029  2032  2035  2039  2042  2045  2048  2051  2054  2057  2060  2063
            2066  2069  2072  2075  2078  2082  2085  2088  2091  2094  2097  2100  2103  2106  2109
            2112  2115  2118  2121  2124  2127  2136  2137  2139  2141  2143  2145  2146  2147  2149
            2151  2153  2155  2158  2190  2191  2195  2196  2197  2198  2202  2282  2283  2288  2289
            2290  2294  2297  2298  2299  2326  2327  2337  2338  2339  2340  2344  2348  2349  2375
            2376  2382  2383  2384  2385  2390  2393  2394  2419  2420  2425  2426  2427  2428  2432
            2435  2436  2437  2511  2512  2522  2523  2524  2525  2529  2532  2533  2534  2607  2608
            2619  2620  2621  2622  2626  2629  2630  2631  2700  2701  2712  2713  2714  2715  2719
            2722  2723  2724  2793  2794  2803  2804  2805  2806  2810  2813  2814  2815  2976  2977
            2991  2992  2993  3003  3004  3014  3015  3016  3017  3021  3024  3025  3026  3042  3048
            3051  3058  3064  3067  3109  3110  3122  3123  3124  3125  3129  3131  3132  3133  3221
            3222  3229  3230  3231  3232  3236  3238  3239  3240  3241  3242  3299  3300  3305  3306
            3307  3308  3312  3314  3315  3316  3317  3318  3384  3385  3390  3391  3392  3393  3397
            3399  3400  3401  3402  3403  3468  3469  3474  3475  3476  3477  3481  3483  3484  3485
            3486  3487  3555  3556  3561  3562  3563  3564  3568  3570  3571  3572  3573  3574  3641
            3642  3647  3648  3649  3650  3654  3729  3730  3735  3736  3737  3738  3742  3744  3745
            3746  3747  3748  3816  3817  3825  3826  3827  3828  3832  3834  3835  3836  3837  3838
            3853  3857  3915  3916  3924  3925  3926  3927  3931  3933  3934  3935  3936  3937  3952
            3956  4014  4015  4023  4024  4025  4026  4030  4032  4033  4034  4035  4036  4051  4055
            4113  4114  4122  4123  4124  4125  4129  4131  4132  4133  4134  4135  4150  4154  4212
            4213  4221  4222  4223  4224  4228  4230  4231  4232  4233  4234  4249  4253  4311  4312
            4320  4321  4322  4323  4327  4329  4330  4331  4332  4333  4348  4352  4410  4411  4419
            4420  4421  4422  4426  4428  4429  4430  4431  4432  4447  4451  4509  4510  4518  4519
            4520  4521  4525  4527  4528  4529  4530  4531  4546  4550  4611  4612  4620  4621  4622
            4623  4627  4629  4630  4631  4632  4633  4735  4736  4744  4745  4746  4747  4751  4753
            4754  4755  4756  4757  4844  4845  4853  4854  4855  4856  4860  4862  4863  4864  4865
            4866  4956  4957  4965  4966  4967  4968  4972  4974  4975  4976  4977  4978  5067  5068
            5076  5077  5078  5079  5083  5085  5086  5087  5088  5089  5178  5179  5187  5188  5189
            5190  5194  5196  5197  5198  5199  5200  5289  5290  5298  5299  5300  5301  5305  5307
            5308  5309  5310  5311  5400  5401  5411  5412  5413  5414  5418  5421  5422  5423  5424
            5425  5510  5511  5518  5519  5520  5521  5525  5527  5528  5529  5530  5531  5570  5571
            5578  5579  5580  5581  5585  5587  5588  5589  5590  5591  5601  5605  5635  5636  5643
            5644  5645  5646  5650  5652  5653  5654  5655  5656  5666  5670  5703  5704  5716  5717
            5718  5719  5723  5725  5726  5727  5728  5729  5820  5821  5833  5834  5835  5836  5840
            5842  5843  5844  5845  5846  5960  5961  5973  5974  5975  5976  5980  5982  5983  5984
            5985  5986  6101  6102  6114  6115  6116  6117  6121  6123  6124  6125  6126  6127  6243
```

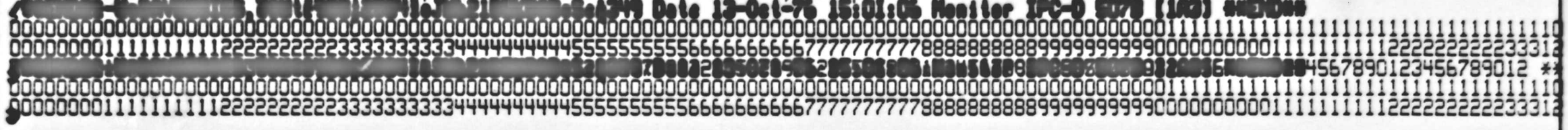
```
                6244    6262    6263    6264    6265    6269    6272    6364    6365    6383    6384    6385    6386    6390    6393
                6488    6491    6493    6494    6496    6499    6505    6508    6509    6513    6515    6517    6519    6525    6526
                6529    6530    6532    6538    6543    6547    6549    6560    6563    6566    6567    6569    6571    6578    6582
                6587    6592    6595    6596    6598    6604    6614    6620    6624    6630    6631    6637    6644    6647    6649
                6696    6770    6849    6918    6927    6930    6932    6942    6943    6944    6945    6946    6947    6948    6949
                6950    6951    6952    6953    6954    6955    6956    6957    6958    6961    6973    6993    6993    6995    6999
                7001
.EQUIV          1268    1269    1271    1292    1293    1294    1295    1296    1297    1298    1299    1300    1301    1302    1331
                1332    1333    1334    1335    1336    1337    1338    1339    1340    1359    1360    1361    1362    1363    1364
                1365    1366    1367    1368    1420    1421    1422    1423    1631    1632    1633    1634    1635    1636    1637
                1638    1639    1640    1641    1642    1643    1644    1645    1646
.EVEN           2158    8593    9042
.IF             1232    1254    1255    1256    1257    1258    1260    1649    1652    1675    1681    1703    1705    1708    1715
                1741    1763    1783    1784    1785    1789    1811    1814    1817    1820    1823    1826    1829    1832    1835
                1838    1841    1844    1847    1850    1853    1856    1859    1862    1865    1868    1871    1874    1877    1880
                1884    1887    1890    1893    1896    1899    1902    1905    1908    1911    1914    1917    1920    1923    1926
                1929    1932    1935    1938    1941    1944    1947    1950    1953    1956    1959    1962    1965    1968    1971
                1974    1977    1980    1983    1986    1989    1994    1997    2000    2003    2006    2009    2012    2015    2018
                2021    2024    2027    2030    2033    2037    2040    2043    2046    2049    2052    2055    2058    2061    2064
                2067    2070    2073    2076    2080    2083    2086    2089    2092    2095    2098    2101    2104    2107    2110
                2113    2116    2119    2122    2125    2131    2136    2137    2139    2141    2143    2145    2146    2147    2149
                2152    2153    2154    2157    2189    2191    2195    2197    2198    2200    2281    2283    2288    2290    2292
                2297    2298    2299    2325    2327    2337    2339    2340    2342    2347    2348    2349    2374    2376    2382
                2384    2385    2388    2393    2394    2418    2420    2425    2427    2428    2430    2435    2436    2510    2512
                2522    2524    2525    2527    2532    2533    2606    2608    2619    2621    2622    2624    2629    2630    2699
                2701    2712    2714    2715    2717    2722    2723    2792    2794    2803    2805    2806    2808    2813    2814
                2975    2977    2991    2993    3002    3004    3014    3016    3017    3019    3024    3025    3041    3048    3051
                3057    3064    3067    3108    3110    3122    3124    3125    3127    3131    3132    3133    3220    3222    3229
                3231    3232    3234    3238    3239    3240    3241    3298    3300    3305    3307    3308    3310    3314    3315
                3316    3317    3383    3385    3390    3392    3393    3395    3399    3400    3401    3402    3467    3469    3474
                3476    3477    3479    3483    3484    3485    3486    3554    3556    3561    3563    3564    3566    3570    3571
                3572    3573    3640    3642    3647    3649    3650    3652    3728    3730    3735    3737    3738    3740    3744
                3745    3746    3747    3815    3817    3825    3827    3828    3830    3834    3835    3836    3837    3853    3856
                3914    3916    3924    3926    3927    3929    3933    3934    3935    3936    3952    3955    4013    4015    4023
                4025    4026    4028    4032    4033    4034    4035    4051    4054    4112    4114    4122    4124    4125    4127
                4131    4132    4133    4134    4150    4153    4211    4213    4221    4223    4224    4226    4230    4231    4232
                4233    4249    4252    4310    4312    4320    4322    4323    4325    4329    4330    4331    4332    4348    4351
                4409    4411    4419    4421    4422    4424    4428    4429    4430    4431    4447    4450    4508    4510    4518
                4520    4521    4523    4527    4528    4529    4530    4546    4549    4610    4612    4620    4622    4623    4625
                4629    4630    4631    4632    4734    4736    4744    4746    4747    4749    4753    4754    4755    4756    4843
                4845    4853    4855    4856    4858    4862    4863    4864    4865    4955    4957    4965    4967    4968    4970
                4974    4975    4976    4977    5066    5068    5076    5078    5079    5081    5085    5086    5087    5088    5177
                5179    5187    5189    5190    5192    5196    5197    5198    5199    5288    5290    5298    5300    5301    5303
                5307    5308    5309    5310    5399    5401    5411    5413    5414    5416    5421    5422    5423    5424    5509
                5511    5518    5520    5521    5523    5527    5528    5529    5530    5569    5571    5578    5580    5581    5583
                5587    5588    5589    5590    5601    5604    5634    5636    5643    5645    5646    5648    5652    5653    5654
                5655    5666    5669    5702    5704    5716    5718    5719    5721    5725    5726    5727    5728    5819    5821
                5833    5835    5836    5838    5842    5843    5844    5845    5959    5961    5973    5975    5976    5978    5982
                5983    5984    5985    6100    6102    6114    6116    6117    6119    6123    6124    6125    6126    6242    6244
                6262    6264    6265    6267    6272    6363    6365    6383    6385    6386    6388    6393    6487    6491    6492
                6493    6494    6495    6496    6498    6504    6507    6509    6513    6515    6517    6525    6526    6531    6537
                6542    6547    6559    6561    6562    6563    6567    6568    6569    6578    6580    6589    6594    6595    6596
                6597    6603    6614    6617    6620    6627    6629    6630    6631    6637    6644    6647    6648    6695    6769
                6848    6917    6926    6930    6932    6933    6943    6944    6945    6946    6947    6948    6950    6951    6952
                6953    6954    6955    6956    6957    6958    6960    6973    6983    6991    6993    6999    7000
.IFF             1254    1256    1257    1258    1682    1704    1706    1709    1715    1741    1790    1811    1816    1819    1822
```

```
        1825  1828  1831  1834  1837  1840  1843  1844  1847  1852  1855  1858  1861  1864  1867
        1870  1873  1876  1879  1882  1886  1889  1892  1895  1898  1901  1904  1907  1910  1913
        1916  1919  1922  1925  1928  1931  1934  1937  1940  1943  1944  1947  1950  1953  1956
        1959  1962  1965  1968  1971  1974  1977  1980  1983  1986  1989  1994  1997  2000  2005
        2008  2011  2014  2017  2020  2023  2026  2029  2032  2035  2039  2042  2045  2048  2049
        2052  2055  2058  2061  2064  2069  2072  2073  2076  2080  2083  2C96  2089  2092  2097
        2100  2101  2104  2107  2110  2115  2118  2121  2124  2125  2136  2152  2154  2190  2191
        2196  2197  2202  2282  2283  2289  2290  2294  2326  2327  2338  2339  2344  2375  2376
        2383  2384  2390  2419  2420  2426  2427  2432  2511  2512  2523  2524  2529  2607  2608
        2620  2621  2626  2700  2701  2713  2714  2719  2793  2794  2804  2805  2810  2976  2977
        2992  2993  3003  3004  3015  3016  3021  3041  3057  3109  3110  3123  3124  3129  3221
        3222  3230  3231  3236  3299  3300  3306  3307  3312  3384  3385  3391  3392  3397  3468
        3469  3475  3476  3481  3555  3556  3562  3563  3568  3641  3642  3648  3649  3654  3729
        3730  3736  3737  3742  3816  3817  3826  3827  3832  3915  3916  3925  3926  3931  4014
        4015  4024  4025  4030  4113  4114  4123  4124  4129  4212  4213  4222  4223  4228  4311
        4312  4321  4322  4327  4410  4411  4420  4421  4426  4509  4510  4519  4520  4525  4611
        4612  4621  4622  4627  4735  4736  4745  4746  4751  4844  4845  4854  4855  4860  4956
        4957  4966  4967  4972  5067  5068  5077  5078  5083  5178  5179  5188  5189  5194  5289
        5290  5299  5300  5305  5400  5401  5412  5413  5418  5510  5511  5519  5520  5525  5570
        5571  5579  5580  5585  5635  5636  5644  5645  5650  5703  5704  5717  5718  5723  5820
        5821  5834  5835  5840  5960  5961  5974  5975  5980  6101  6102  6115  6116  6121  6243
        6244  6263  6264  6269  6364  6365  6394  6385  6390  6488  6495  6499  6504  6507  6526
        6532  6560  6563  6569  6595  6598  6603  6620  6631  6644  6649  6696  6770  6849  6918
        6927  6961  6991  7001
.IFT    2158  6577  6630
.IFTF   2158  6575  6629
.IIF    1231  1236  1241  1242  1251  1252  1253  1254  1257  1258  1259  1675  1789  2137  2139
        2145  2146  2147  2149  2150  2153  6493  6499  6500  6511  6526  6530  6538  6539  6540
        6541  6542  6543  6576  6577  6592  6595  6596  6604  6605  6606  6607  6608  6609  6631
        6647  6768  6942  6943  6944  6945  6946  6947  6948  6950  6951  6952  6953  6954  6955
        6956  6957  6958
.IRP    1652  1810  1883  1944  1993  2036  2079  2189  2281  2325  2374  2418  2510  2606  2699
        2792  2975  3002  3108  3220  3298  3383  3467  3554  3640  3728  3815  3914  4013  4112
        4211  4310  4409  4508  4610  4734  4843  4955  5066  5177  5288  5399  5509  5569  5634
        5702  5819  5959  6100  6242  6363  6515  6644  6667  6687  6862  6902  6967  6983
.LIST      1  1184  1257  1650  1652  1675  1741  1743  1744  1745  1746  1747  1748  1749  1750
        1751  1752  1753  1754  1755  1756  1757  1758  1759  1760  1761  1762  1763  1764  1765
        1766  1767  1768  1769  1770  1771  1772  1773  1774  1775  1776  1777  1778  1779  1780
        1781  1782  1783  2153  2158  2189  2197  2281  2290  2325  2339  2374  2384  2418  2427
        2510  2524  2606  2621  2699  2714  2792  2805  2975  2993  3002  3016  3108  3124  3220
        3231  3298  3307  3383  3392  3467  3476  3554  3563  3640  3649  3728  3737  3815  3827
        3914  3926  4013  4025  4112  4124  4211  4223  4310  4322  4409  4421  4508  4520  4610
        4622  4734  4746  4843  4855  4955  4967  5066  5078  5177  5189  5288  5300  5399  5413
        5509  5520  5569  5580  5634  5645  5702  5718  5819  5835  5959  5975  6100  6116  6242
        6264  6363  6385  6499  6515  6517  6542  6631  6932  6933  6942  6943  6944  6945  6946
        6947  6948  6949  6950  6951  6952  6953  6954  6955  6956  6957  6958  6959
.MACRO     1  1188  1189  1190  1192  1194  1195  1196  1197  1198  1199  1200  1201  1203  1204
        1205  1206  1207  1208  1209  1210  1211  1212  1213  1214  1215  1216  1217  1218  1219
        1220  1221  1222  1223  1224  1225  1258  1708  2188  2281  2325  2373  2417  2510  2605
        2606  2699  2792  2975  3002  3108  3217  3219  3298  3383  3467  3554  3640  3728  3814
        3815  3914  4013  4112  4211  4310  4409  4508  4607  4608  4609  4610  4734  4843  4955
        5066  5177  5288  5399  5509  5567  5569  5634  5699  5700  5701  5702  5819  5959  6100
        6242  6363  6933  8402
.MCALL  1228  1229  1230  1650
.NLIST     1  1185  1257  1650  1652  1675  1741  1743  1744  1745  1746  1747  1748  1749  1750
        1751  1752  1753  1754  1755  1756  1757  1758  1759  1760  1761  1762  1763  1764  1765
```

```
            1766    1767    1768    1769    1770    1771    1772    1773    1774    1775    1776    1777    1778    1779    1780
            1781    1782    1783    2153    2158    2189    2197    2281    2290    2325    2339    2374    2384    2418    2427
            2510    2524    2606    2621    2699    2714    2792    2805    2975    2993    3002    3016    3108    3124    3220
            3231    3298    3307    3383    3392    3467    3476    3554    3563    3640    3649    3728    3737    3815    3827
            3914    3926    4013    4025    4112    4124    4211    4223    4310    4322    4409    4421    4508    4520    4610
            4622    4734    4746    4843    4855    4955    4967    5066    5078    5177    5189    5288    5300    5399    5413
            5509    5520    5569    5580    5634    5645    5702    5718    5819    5835    5959    5975    6100    6116    6242
            6264    6363    6385    6499    6515    6517    6542    6631    6932    6933    6942    6943    6944    6945    6946
            6947    6948    6949    6950    6951    6952    6953    6954    6955    6956    6957    6958    6959
.PAGE       1708    1789
.REM           1
.REPT       1675    1743    1763
.SBTTL      1247    1261    1386    1397    1411    1560    1669    1676    1683    1710    1791    2189    2281    2325    2374
            2418    2510    2606    2699    2792    2975    3002    3108    3220    3298    3383    3467    3554    3640    3728
            3815    3914    4013    4112    4211    4310    4409    4508    4610    4734    4843    4955    5066    5177    5288
            5399    5509    5569    5634    5702    5819    5959    6100    6242    6363    6489    6533    6599    6650    6697
            6771    6850    6919    6934    6962    7002
.TITLE      1231
.WORD       1675    1703    1705    1718    1721    1722    1723    1724    1727    1728    1729    1730    1731    1732    1741
            1743    1744    1745    1746    1747    1748    1749    1750    1751    1752    1753    1754    1755    1756    1757
            1758    1759    1760    1761    1762    1763    1764    1765    1766    1767    1768    1769    1770    1771    1772
            1773    1774    1775    1776    1777    1778    1779    1780    1781    1782    1811    1814    1817    1820    1823
            1826    1829    1832    1835    1838    1841    1844    1847    1850    1853    1856    1859    1862    1865    1868
            1871    1874    1877    1880    1884    1887    1890    1893    1896    1899    1902    1905    1908    1911    1914
            1917    1920    1923    1926    1929    1932    1935    1938    1941    1945    1948    1951    1954    1957    1960
            1963    1966    1969    1972    1975    1978    1981    1984    1987    1990    1994    1997    2000    2003    2006
            2009    2012    2015    2018    2021    2024    2027    2030    2033    2037    2040    2043    2046    2049    2052
            2055    2058    2061    2064    2067    2070    2073    2076    2080    2083    2086    2089    2092    2095    2098
            2101    2104    2107    2110    2113    2116    2119    2122    2125    2225    2480    2578    2693    2694    2786
            2787    2934    2936    2938    2941    2942    2943    2944    2945    2946    2947    2948    2949    2950    2952
            2953    2954    2955    2956    2957    2959    2960    2961    2962    2963    2964    3103    3104    6332    6333
            6453    6454    6504    6507    6765    6846    6992    6994    7079    7098    7118    7125    7132    7139    7146
            7151    7152    7153    7154    7155    7156    7157    7158    7159    7160    7161    7162    7196    7208    7215
            7217    7270    7285    7287    7291    7293    7317    7337    7352    7358    7369    7375    7376    7416    8594
            9046    9049    9052    9055    9068    9071    9074    9077    9092    9095    9098    9101    9106    9109    9111
            9113    9115    9117    9120    9123    9128    9132    9135    9141    9144    9146    9153    9157    9159    9162
            9165


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*DEKBCB,DEKBCB.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:SYSMAC.SML,DSKM:DEKBCB.P11
RUN-TIME: 68 91 16 SECONDS
RUN-TIME RATIO: 208/176=1.1
CORE USED:  36K  (71 PAGES)
```



M01

```
Spooler runtime 30 Seconds, 126 KCS, 882 disk reads, 6 disk writes, 218 pages
[redacted] Date 13-Oct-76 15:01:06 Monitor IPC-D 507B [1A0] *END**
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001111111111111111111111111111111110
00000000011111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111111222222222233312
[redacted]4567890123456789012 **
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001111111111111111111111111111111110
00000000011111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111111222222222233312
```