

11/70-74

11/70-74 CACHE # 1  
CEKBCC0

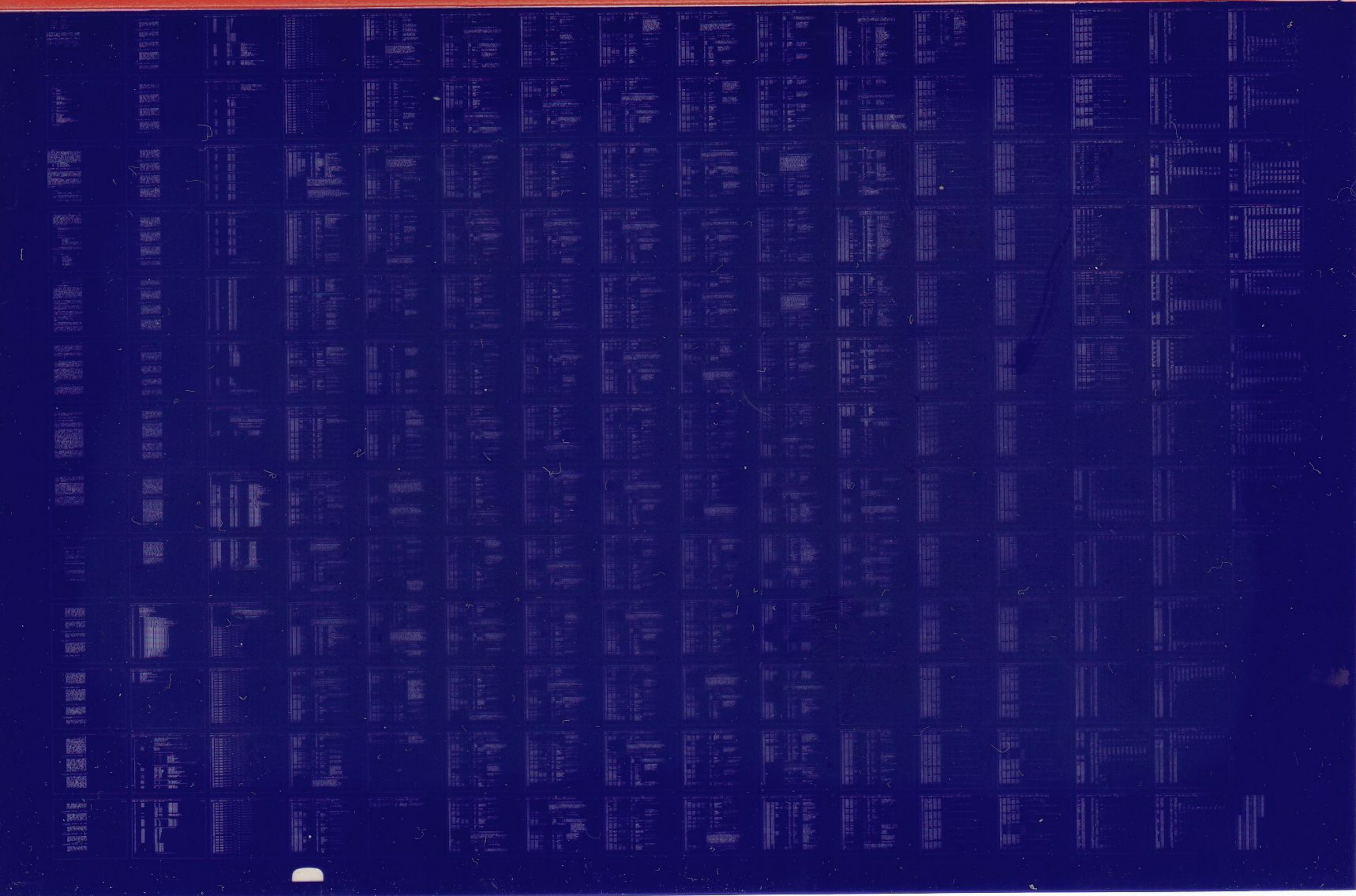
AH-0010C-MC

COPYRIGHT 75-80  
FICHE 1 OF 1

JAN 1980

**digital**

MADE IN USA





## IDENTIFICATION

SEQ 0001

PRODUCT CODE: AC-0009C-MC  
PRODUCT NAME: CEKBCCO PDP-11/70-74MP CACHE DIAGNOSTIC PART 1  
DATE CREATED: MAY, 1979  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: ANTHONY VEZZA  
MODIFIED BY: ERNEST PREISIG

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975, 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	DECX/11

## CONTENTS

1. ABSTRACT
2. REQUIREMENTS
  - 2.1 EQUIPMENT
  - 2.2 STORAGE
  - 2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
  - 3.1 METHOD
4. STARTING PROCEDURE
  - 4.1 CONTROL SWITCH SETTINGS
  - 4.2 STARTING ADDRESS
  - 4.3 PROGRAM AND OPERATOR ACTION
  - 4.4 SPECIAL OPERATOR INTERVENTION OPTIONS
5. OPERATING PROCEDURE
  - 5.1 OPERATIONAL SWITCH SETTINGS
  - 5.2 SUBROUTINE ABSTRACTS
  - 5.3 OPERATOR ACTION
6. ERRORS
  - 6.1 ERROR HALTS AND DESCRIPTION
  - 6.2 ERROR RECOVERY
7. RESTRICTIONS
  - 7.1 STARTING RESTRICTIONS
  - 7.2 OPERATING RESTRICTIONS
8. MISCELLANEOUS
  - 8.1 EXECUTION TIME
  - 8.2 STACK POINTER
  - 8.3 PASS COUNT
  - 8.4 ITERATIONS
  - 8.5 OSCILLOSCOPE SYNC POINTS
  - 8.6 RESTORING LOADER OR MONITOR
  - 8.7 OPTIONAL POWER DOWN POWER UP TEST
  - 8.8 MEMORY MANAGEMENT RESTRICTIONS/OPTIONS
  - 8.9 CRITICAL DEPENDENCE OF SOME TESTS ON THE CACHE REGISTERS
9. PROGRAM DESCRIPTION
  - 9.1 CEKBC
10. LISTINGS
  - 10.1 CEKBC

## 1. ABSTRACT

THE PROGRAMS, CEKBC AND CEKBD, ARE INTENDED TO BE USED AS AIDS FOR THE REPAIR AND MAINTENANCE OF THE CACHE MEMORY SYSTEM IN THE PDP 11/70-74MP 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 A 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 CEKBC IS DESIGNED TO TEST THE FIRST TWO OF THESE BOARDS, WHILE CEKBD 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 CEKBC WITHOUT ERROR DOES NOT RULE OUT A FAULTY COMPONENT ON THE CCB (CACHE CONTROL) BOARD.

TESTING HAS BEEN DIVIDED INTO TWO PROGRAMS ONLY BECAUSE OF THE RESTRICTIONS OF CORE SIZE RATHER THAN TO PROVIDE A MEANS OF TESTING TWO OF THE BOARDS WITH ONE PROGRAM AND THE OTHER TWO BOARDS WITH A SECOND PROGRAM. NOTE THAT CEKBD IS DESIGNED TO RUN AFTER CEKBC. IF THIS HIERARCHY IS NOT HEEDED, THAT IS IF CEKBD IS RUN BEFORE CEKBC, THEN THE ERROR REPORTING FROM CEKBD SHOULD NOT BE STRICTLY INTERPRETED.

THIS DIAGNOSTIC SUPPORTS THE KB11-B/C, AND KB11-CM PROCESSORS.

## 2. REQUIREMENTS

2.1 EQUIPMENT - PDP 11/70-74MP CPU WITH OPERATORS CONSOLE LA30 OR EQUIVALENT TERMINAL.

2.2 STORAGE - BOTH PROGRAMS, CEKBC AND CEKBD, 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 ASSUMES 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, CEKBC AND CEKBD, 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 ORDER: CEKBC BEFORE CEKBD. IN FACT CEKBD ASSUMES THAT MUCH OF WHAT IS TESTED IN CEKBC IS OPERATIONAL FOR DOING ITS FAULT ANALYSIS.

### 3. LOADING PROCEDURE

3.1 METHOD - BOTH CEKBC AND CEKBD ARE LOADED FROM THE XXDP MEDIA. REFER TO THE XXDP MANUAL FOR FURTHER INFORMATION.

### 4. STARTING PROCEDURE

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 12 OF THE SWITCH REGISTER IS ON, THEN CEKBD 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 CEKBC:

SW<15>=1	HALT ON ERROR
SW<14>=1	LOOP ON TEST
SW<13>=1	INHIBIT ERROR TYPINGS
SW<12>	NOT USED IN CEKBC
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<6:0>
SW<7>=1	SKIP EXECUTION OF TESTS WHICH USE MEMORY MANAGEMENT.
SW<6:0>	TEST NUMBER FOR LOOPING WHEN SW<8>=1

CEKBD USES THE SAME SWITCH SETTINGS AS CEKBC EXCEPT:

SW<12> -1        RUN THE OPERATOR INTERVENTION NEEDED  
                  POWER UP TEST

5.2        SUBROUTINE ABSTRACTS - BOTH CEKBC AND CEKBD  
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,  
LOADRS AND HIADRS. IT THEN GIVES CONTROL TO THE  
TEST FOLLOWING THE ONE DURING WHICH THE ERROR  
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 SETING  
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 INSTRUCTION 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 ASCII 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 CEKBD 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 CEKBC, THE MAINTENANCE REGISTER COUNT PATTERN TEST, HALTS THE PROCESSOR IN THE SITUATION WHERE IT CAN'T CLEAR THE MAINTENANCE REGISTER. HERE PROCEEDING 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 THE 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 .SEOP.



## 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 LOCATION 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 OVERRIDEN BY A ONE (1) MAKING ITERATIONS MEANINGLESS ON THAT FIRST PASS.

8.5 OSCILLOSCOPE SYNC POINTS - WHENEVER POSSIBLE EACH TEST HAS BEEN GIVEN AN OSCILLOSCOPE SYNC POINT (A NOP INSTRUCTION). THE ADDRESS OF THE CONDITION 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 A NOP INSTRUCTION IN CRUCIAL PARTS OF A TEST, THE USER WILL HAVE A VERY CONVENIENT SYNC FOR MANY SIGNALS HE MAY WISH TO OBSERVE. 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 RESTORES 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 (CEKBD ONLY. SEE 5.1).

8.8 MEMORY MANAGEMENT RESTRICTIONS/OPTIONS - MANY OF THE TESTS REQUIRE THE USE OF EXTENSIVE MEMORY MANAGEMENT MAPPING FACILITIES. THESE TESTS MUST ASSUME THE MEMORY MANAGEMENT (AND SOME OF 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 IMPLEMENTED 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.

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!

9.1 CEKBC

COPYRIGHT 1975,1979 DIGITAL EQUIPMENT CORPORATION MAYNARD, MASS. 01754

COPYRIGHT (C) 1975, 1979 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-A5-1).

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. IF THIS IS A KB11-CM CPU THEN BITS 9, 11, 13, AND 14 ARE ALSO TESTED.

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

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 TESTS THROUGH TEST10, 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.

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 SETTABLE 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 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.

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  
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 ADDRESSES, 17777776, WILL BE USED TO 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

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.

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 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.

## 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

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!

20	OPERATIONAL SWITCH SETTINGS
34	BASIC DEFINITIONS
159	CACHE REGISTER DEFINITIONS
170	CPU REGISTER DEFINITIONS
184	MEMORY MANAGEMENT DEFINITIONS
333	UNIBUS MAP REGISTER DEFINITIONS
442	TRAP CATCHER
449	STARTING ADDRESS(ES)
456	ACT11 HOOKS
483	COMMON TAGS
571	ERROR POINTER TABLE
1099	T1 CACHE REGISTERS RESPONSE TEST
1192	T2 CACHE REGISTERS DATA PATH, READ ZEROES TEST
1243	T3 CACHE REGISTERS DATA PATH, READ ONES TEST
1293	T4 CACHE CONTROL REGISTER COUNT PATTERN TEST
1365	T5 CACHE HIT/MISS AND CONTROL REGISTER SIMPLE MISSES TEST
1458	T6 CACHE HIT/MISS AND CONTROL REGISTER SIMPLE HIT TEST
1555	T7 CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 0 TEST
1649	T10 CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 1 TEST
1743	T11 CACHE HIT/MISS REGISTER PATTERNS TEST
1927	T12 CACHE CONTROL AND HIT/MISS REGISTERS EVALUATION ROUTINE
1955	T13 CACHE CONTROL LOGIC, 'RANDOM' FLIP FLOP TEST
2062	T14 CACHE MAINTENANCE REGISTER COUNT PATTERN TEST
2175	T15 CACHE MAINTENANCE AND ERROR REGISTERS TEST 1
2254	T16 CACHE MAINTENANCE AND ERROR REGISTERS TEST 2
2340	T17 CACHE MAINTENANCE AND ERROR REGISTERS TEST 3
2425	T20 CACHE MAINTENANCE AND ERROR REGISTERS TEST 4
2513	T21 CACHE MAINTENANCE AND ERROR REGISTERS TEST 5
2600	T22 CACHE MAINTENANCE AND ERROR REGISTERS TEST 6
2689	T23 CACHE MAINTENANCE AND ERROR REGISTERS TEST 7
2777	T24 CACHE MAINTENANCE AND ERROR REGISTERS TEST 10
2877	T25 CACHE MAINTENANCE AND ERROR REGISTERS TEST 11
2977	T26 CACHE MAINTENANCE AND ERROR REGISTERS TEST 12
3077	T27 CACHE MAINTENANCE AND ERROR REGISTERS TEST 13
3177	T30 CACHE MAINTENANCE AND ERROR REGISTERS TEST 14
3277	T31 CACHE MAINTENANCE AND ERROR REGISTERS TEST 15
3377	T32 CACHE MAINTENANCE AND ERROR REGISTERS TEST 16
3477	T33 CACHE MAINTENANCE AND ERROR REGISTERS TEST 17
3580	T34 CACHE MAINTENANCE AND ERROR REGISTERS TEST 20
3705	T35 CACHE MAINTENANCE AND ERROR REGISTERS TEST 21
3815	T36 CACHE MAINTENANCE AND ERROR REGISTERS TEST 22
3928	T37 CACHE MAINTENANCE AND ERROR REGISTERS TEST 23
4040	T40 CACHE MAINTENANCE AND ERROR REGISTERS TEST 24
4152	T41 CACHE MAINTENANCE AND ERROR REGISTERS TEST 25
4264	T42 CACHE MAINTENANCE AND ERROR REGISTERS TEST 26
4376	T43 CACHE ERROR REGISTER UNIBUS TIME OUT TEST
4487	T44 CACHE CONTROL REGISTER DISABLE TRAPS TEST 1
4548	T45 CACHE CONTROL REGISTER DISABLE TRAPS TEST 2
4614	T46 CACHE CONTROL REGISTER DISABLE TRAPS TEST 3
4683	T47 CACHE ERROR REGISTER LOCK UP TEST 1
4801	T50 CACHE ERROR REGISTER LOCK UP TEST 2
4942	T51 CACHE ERROR REGISTER LOCK UP TEST 3
5084	T52 CACHE ERROR REGISTER LOCK UP TEST 4
5227	T53 MAIN MEMORY DATA PARITY CHECKERS LOW BYTE TEST
5349	T54 MAIN MEMORY DATA PARITY CHECKERS HIGH BYTE TEST
5476	END OF PASS ROUTINE



5520	SCOPE HANDLER ROUTINE
5588	ERROR HANDLER ROUTINE
5639	SAVE AND RESTORE R0-R5 ROUTINES
5686	TYPE ROUTINE
5760	BINARY TO OCTAL (ASCII) AND TYPE
5839	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
5908	TRAP DECODER
5923	TRAP TABLE
5951	POWER DOWN AND UP ROUTINES
5990	ROUTINE TO SIZE MEMORY
6087	DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56

```

.TITLE CEKBC-C PDP 11/70-74MP CACHE DIAGNOSTIC PART 1
.*COPYRIGHT (C) 1975, 1978
.*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-A5-1).
.*
$TN=1
$SWR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
$SWR=167400
$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
000700 SUPSTK= STACK-200 ;;SUPERVISOR STACK
000600 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
    
```

```

57      000003      R3=      %3          ;;GENERAL REGISTER
58      000004      R4=      %4          ;;GENERAL REGISTER
59      000005      R5=      %5          ;;GENERAL REGISTER
60      000006      R6=      %6          ;;GENERAL REGISTER
61      000007      R7=      %7          ;;GENERAL REGISTER
62      .EQUIV      R0,R10      ;;GENERAL REGISTER
63      .EQUIV      R1,R11      ;;GENERAL REGISTER
64      .EQUIV      R2,R12      ;;GENERAL REGISTER
65      .EQUIV      R3,R13      ;;GENERAL REGISTER
66      .EQUIV      R4,R14      ;;GENERAL REGISTER
67      .EQUIV      R5,R15      ;;GENERAL REGISTER
68      000006      SP=%6          ;;KERNEL STACK POINTER
69      .EQUIV      SP,KSP      ;;SUPERVISOR STACK POINTER
70      .EQUIV      SP,SSP      ;;USER STACK POINTER
71      .EQUIV      SP,USP
72      000007      PC=%7
73
74      ;*PRIORITY LEVEL DEFINITIONS
75      000000      PR0=      0          ;;PRIORITY LEVEL 0
76      000040      PR1=      40         ;;PRIORITY LEVEL 1
77      000100      PR2=      100        ;;PRIORITY LEVEL 2
78      000140      PR3=      140        ;;PRIORITY LEVEL 3
79      000200      PR4=      200        ;;PRIORITY LEVEL 4
80      000240      PR5=      240        ;;PRIORITY LEVEL 5
81      000300      PR6=      300        ;;PRIORITY LEVEL 6
82      000340      PR7=      340        ;;PRIORITY LEVEL 7
83
84      ;*'SWITCH REGISTER' SWITCH DEFINITIONS
85      1J0000      SW15=     100000
86      040000      SW14=     40000
87      020000      SW13=     20000
88      010000      SW12=     10000
89      004000      SW11=     4000
90      002000      SW10=     2000
91      001000      SW09=     1000
92      000400      SW08=     400
93      000200      SW07=     200
94      000100      SW06=     100
95      000040      SW05=     40
96      000020      SW04=     20
97      000010      SW03=     10
98      000004      SW02=     4
99      000002      SW01=     2
100     000001      SW00=     1
101     .EQUIV      SW09,SW9
102     .EQUIV      SW08,SW8
103     .EQUIV      SW07,SW7
104     .EQUIV      SW06,SW6
105     .EQUIV      SW05,SW5
106     .EQUIV      SW04,SW4
107     .EQUIV      SW03,SW3
108     .EQUIV      SW02,SW2
109     .EQUIV      SW01,SW1
110     .EQUIV      SW00,SW0
111
112     ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
    
```

```

113      100000      BIT15= 100000
114      040000      BIT14= 40000
115      020000      BIT13= 20000
116      010000      BIT12= 10000
117      004000      BIT11= 4000
118      002000      BIT10= 2000
119      001000      BIT09= 1000
120      000400      BIT08= 400
121      000200      BIT07= 200
122      000100      BIT06= 100
123      000040      BIT05= 40
124      000020      BIT04= 20
125      000010      BIT03= 10
126      000004      BIT02= 4
127      000002      BIT01= 2
128      000001      BIT00= 1
129      .EQUIV      BIT09,BIT9
130      .EQUIV      BIT08,BIT8
131      .EQUIV      BIT07,BIT7
132      .EQUIV      BIT06,BIT6
133      .EQUIV      BIT05,BIT5
134      .EQUIV      BIT04,BIT4
135      .EQUIV      BIT03,BIT3
136      .EQUIV      BIT02,BIT2
137      .EQUIV      BIT01,BIT1
138      .EQUIV      BIT00,BIT0
    
```

```

140      ;*BASIC 'CPU' TRAP VECTOR ADDRESSES
141      000004      ERRVEC= 4          ;;TIME OUT AND OTHER ERRORS
142      000010      RESVEC= 10         ;;RESERVED AND ILLEGAL INSTRUCTIONS
143      000014      TBITVEC=14         ;;'T' BIT
144      000014      TRTVEC= 14         ;;TRACE TRAP
145      000014      BPTVEC= 14         ;;BREAKPOINT TRAP (BPT)
146      000020      IOTVEC= 20         ;;INPUT/OUTPUT TRAP (IO*) **SCOPE**
147      000024      PWRVEC= 24         ;;POWER FAIL
148      000030      EMTVEC= 30         ;;EMULATOR TRAP (EMT) **ERROR**
149      000034      TRAPVEC=34         ;;'TRAP' TRAP
150      000060      TKVEC= 60          ;;TTY KEYBOARD VECTOR
151      000064      TPVEC= 64          ;;TTY PRINTER VECTOR
152      000114      CACHVEC=114        ;;CACHE ERROR INTERRUPT VECTOR
153      000240      PIRQVEC=240        ;;PROGRAM INTERRUPT REQUEST VECTOR
154      000250      MMVEC= 250         ;;MEMORY MANAGEMENT VECTOR
    
```

.SBTTL CACHE REGISTER DEFINITIONS

```

158      177740      LOADRS = 177740    ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
159      177742      HIADRS = 177742    ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
160      177744      MEMERR = 177744    ;;CACHE ERROR REGISTER
161      177746      CONTRL = 177746    ;;MEMORY CONTROL REGISTER
162      177750      MAINT = 177750     ;;MEMORY MAINTENANCE REGISTER
163      177752      HITMIS = 177752    ;;HIT MISS REGISTER '1' IMPLIES HIT IN CACHE
    
```

.SBTTL CPU REGISTER DEFINITIONS

167  
168

169  
170 177760 SIZELO = 177760 ::MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR  
171 ::TO GET TO THE LAST 32 WORDS OF MEMORY  
172 177762 SIZEHI = 177762 ::HIGH SIZE REGISTER, RESERVED FOR FUTURE USE  
173 ::CURRENTLY ALL ZERO  
174 177764 SYSTID = 177764 ::SYSTEM ID REGISTER  
175 177766 CPUERR = 177766 ::CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED  
176 ::THE TRAP TO ERRVEC (000004)  
177  
178  
179

180  
181 .SBT'L MEMORY MANAGEMENT DEFINITIONS  
182

183  
184 ;\*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES  
185

186 177572 MMR0= 177572  
187 177574 MMR1= 177574  
188 177576 MMR2= 177576  
189 172516 MMR3= 172516  
190 .EQUIV MMR0,SRO  
191 .EQUIV MMR1,SR1  
192 .EQUIV MMR2,SR2  
193 .EQUIV MMR3,SR3  
194

195 ;\*USER 'I' PAGE DESCRIPTOR REGISTERS  
196

197 177600 UIPDR0= 177600  
198 177602 UIPDR1= 177602  
199 177604 UIPDR2= 177604  
200 177606 UIPDR3= 177606  
201 177610 UIPDR4= 177610  
202 177612 UIPDR5= 177612  
203 177614 UIPDR6= 177614  
204 177616 UIPDR7= 177616  
205

206 ;\*USER 'D' PAGE DESCRIPTOR REGISTORS  
207

208 177620 UDPDR0= 177620  
209 177622 UDPDR1= 177622  
210 177624 UDPDR2= 177624  
211 177626 UDPDR3= 177626  
212 177630 UDPDR4= 177630  
213 177632 UDPDR5= 177632  
214 177634 UDPDR6= 177634  
215 177636 UDPDR7= 177636  
216

217 ;\*USER 'I' PAGE ADDRESS REGISTERS  
218

219 177640 UIPAR0= 177640  
220 177642 UIPAR1= 177642  
221 177644 UIPAR2= 177644  
222 177646 UIPAR3= 177646  
223 177650 UIPAR4= 177650  
224 177652 UIPAR5= 177652

225 177654 UIPAR6= 177654  
226 177656 UIPAR7= 177656  
227  
228 ;\*USER 'D' PAGE ADDRESS REGISTERS  
229  
230 177660 UDPAR0= 177660  
231 177662 UDPAR1= 177662  
232 177664 UDPAR2= 177664  
233 177666 UDPAR3= 177666  
234 177670 UDPAR4= 177670  
235 177672 UDPAR5= 177672  
236 177674 UDPAR6= 177674  
237 177676 UDPAR7= 177676  
238  
239 ;\*SUPERVISOR 'I' PAGE DESCRIPTOR REGISTERS  
240  
241 172200 SIPDR0= 172200  
242 172202 SIPDR1= 172202  
243 172204 SIPDR2= 172204  
244 172206 SIPDR3= 172206  
245 172210 SIPDR4= 172210  
246 172212 SIPDR5= 172212  
247 172214 SIPDR6= 172214  
248 172216 SIPDR7= 172216  
249  
250 ;\*SUPERVISOR 'D' PAGE DESCRIPTOR REGISTERS  
251  
252 172220 SDPDR0= 172220  
253 172222 SDPDR1= 172222  
254 172224 SDPDR2= 172224  
255 172226 SDPDR3= 172226  
256 172230 SDPDR4= 172230  
257 172232 SDPDR5= 172232  
258 172234 SDPDR6= 172234  
259 172236 SDPDR7= 172236  
260  
261 ;\*SUPERVISOR 'I' PAGE ADDRESS REGISTERS  
262  
263 172240 SIPAR0= 172240  
264 172242 SIPAR1= 172242  
265 172244 SIPAR2= 172244  
266 172246 SIPAR3= 172246  
267 172250 SIPAR4= 172250  
268 172252 SIPAR5= 172252  
269 172254 SIPAR6= 172254  
270 172256 SIPAR7= 172256  
271  
272 ;\*SUPERVISOR 'D' PAGE ADDRESS REGISTERS  
273  
274 172260 SDPAR0= 172260  
275 172262 SDPAR1= 172262  
276 172264 SDPAR2= 172264  
277 172266 SDPAR3= 172266  
278 172270 SDPAR4= 172270  
279 172272 SDPAR5= 172272  
280 172274 SDPAR6= 172274

281 172276 SDPAR7= 172276

282  
283 ;\*KERNEL 'I' PAGE DESCRIPTOR REGISTERS

284  
285 172300 KIPDR0= 172300  
286 172302 KIPDR1= 172302  
287 172304 KIPDR2= 172304  
288 172306 KIPDR3= 172306  
289 172310 KIPDR4= 172310  
290 172312 KIPDR5= 172312  
291 172314 KIPDR6= 172314  
292 172316 KIPDR7= 172316

293  
294 ;\*KERNEL 'D' PAGE DESCRIPTOR REGISTERS

295  
296 172320 KDPDR0= 172320  
297 172322 KDPDR1= 172322  
298 172324 KDPDR2= 172324  
299 172326 KDPDR3= 172326  
300 172330 KDPDR4= 172330  
301 172332 KDPDR5= 172332  
302 172334 KDPDR6= 172334  
303 172336 KDPDR7= 172336

304  
305 ;\*KERNEL 'I' PAGE ADDRESS REGISTERS

306  
307 172340 KIPAR0= 172340  
308 172342 KIPAR1= 172342  
309 172344 KIPAR2= 172344  
310 172346 KIPAR3= 172346  
311 172350 KIPAR4= 172350  
312 172352 KIPAR5= 172352  
313 172354 KIPAR6= 172354  
314 172356 KIPAR7= 172356

315  
316 ;\*KERNEL 'D' PAGE ADDRESS REGISTERS

317  
318 172360 KDPAR0= 172360  
319 172362 KDPAR1= 172362  
320 172364 KDPAR2= 172364  
321 172366 KDPAR3= 172366  
322 172370 KDPAR4= 172370  
323 172372 KDPAR5= 172372  
324 172374 KDPAR6= 172374  
325 172376 KDPAR7= 172376

326  
327  
328  
329  
330 .SBTTL UNIBUS MAP REGISTER DEFINITIONS

331  
332 ;\*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'  
333 ;\*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

334  
335  
336



337	170200	MAPL00 = 170200
338	170202	MAPH00 = 170202
339	170204	MAPL01 = 170204
340	170206	MAPH01 = 170206
341	170210	MAPL02 = 170210
342	170212	MAPH02 = 170212
343	170214	MAPL03 = 170214
344	170216	MAPH03 = 170216
345	170220	MAPL04 = 170220
346	170222	MAPH04 = 170222
347	170224	MAPL05 = 170224
348	170226	MAPH05 = 170226
349	170230	MAPL06 = 170230
350	170232	MAPH06 = 170232
351	170234	MAPL07 = 170234
352	170236	MAPH07 = 170236
353	170240	MAPL10 = 170240
354	170242	MAPH10 = 170242
355	170244	MAPL11 = 170244
356	170246	MAPH11 = 170246
357	170250	MAPL12 = 170250
358	170252	MAPH12 = 170252
359	170254	MAPL13 = 170254
360	170256	MAPH13 = 170256
361	170260	MAPL14 = 170260
362	170262	MAPH14 = 170262
363	170264	MAPL15 = 170264
364	170266	MAPH15 = 170266
365	170270	MAPL16 = 170270
366	170272	MAPH16 = 170272
367	170274	MAPL17 = 170274
368	170276	MAPH17 = 170276
369	170300	MAPL20 = 170300
370	170302	MAPH20 = 170302
371	170304	MAPL21 = 170304
372	170306	MAPH21 = 170306
373	170310	MAPL22 = 170310
374	170312	MAPH22 = 170312
375	170314	MAPL23 = 170314
376	170316	MAPH23 = 170316
377	170320	MAPL24 = 170320
378	170320	MAPH24 = 170320
379	170324	MAPL25 = 170324
380	170326	MAPH25 = 170326
381	170330	MAPL26 = 170330
382	170332	MAPH26 = 170332
383	170334	MAPL27 = 170334
384	170336	MAPH27 = 170336
385	170340	MAPL30 = 170340
386	170342	MAPH30 = 170342
387	170344	MAPL31 = 170344
388	170346	MAPH31 = 170346
389	170350	MAPL32 = 170350
390	170352	MAPH32 = 170352
391	170354	MAPL33 = 170354
392	170356	MAPH33 = 170356

```
393      170360      MAPL34 = 170360
394      170362      MAPH34 = 170362
395      170364      MAPL35 = 170364
396      170366      MAPH35 = 170366
397      170370      MAPL36 = 170370
398      170372      MAPH36 = 170372
399      170374      MAPL37 = 170374
400      170376      MAPH37 = 170376
401      .EQUIV      MAPL00,MAPL0
402      .EQUIV      MAPH00,MAPH0
403      .EQUIV      MAPL01,MAPL1
404      .EQUIV      MAPH01,MAPH1
405      .EQUIV      MAPL02,MAPL2
406      .EQUIV      MAPH02,MAPH2
407      .EQUIV      MAPL03,MAPL3
408      .EQUIV      MAPH03,MAPH3
409      .EQUIV      MAPL04,MAPL4
410      .EQUIV      MAPH04,MAPH4
411      .EQUIV      MAPL05,MAPL5
412      .EQUIV      MAPH05,MAPH5
413      .EQUIV      MAPL06,MAPL6
414      .EQUIV      MAPH06,MAPH6
415      .EQUIV      MAPL07,MAPL7
416      .EQUIV      MAPH07,MAPH7
417
418
419
420
421
422
423
424
425
426
427
428      000011      TAB 11
429      000044      S1M0 44
430      000030      SOM1-30
431      000054      S1MOM1 54
432      000034      SOMOM1-34
433      000014      M*MC=14
434      000014      MOM1=M1M0
435      140000      TESTR1=140000
436      142000      TESTR2=142000
437      144000      TESTR3=144000
438
439      .SBTTL      TRAP CATCHER
440
441      000000      . = 0
442      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
443      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
444      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
445
446      .SBTTL      STARTING ADDRESS(ES)
447      000200      .-200
448
```

```
449 000200 000137 003014          JMP      @START          ;; JUMP TO STARTING ADDRESS OF PROGRAM
450
451                               ;:*****
452
453          .SBITL          ACT11 HOOKS
454
455          ;*THE FOLLOWING LOCATIONS ARE SFTUP TO BE USED WITH ACT11
456          ;*
457          ;*LOCATION 46 WILL CONTAIN THE ADDRESS OF THE LOGICAL
458          ;*END OF THE PROGRAM.
459          ;*LOCATION 52 IS USED TO SPECIFY PROGRAM OPERATING REQUIREMENTS
460          ;*AND/OR RESTRICTIONS. THIS IS ACCOMPLISHED BY SETTING VARIOUS BITS
461          ;*TO A ONE OR A ZERO. THE BITS USED AND THERE MEANING ARE:
462          ;*
463          ;*      BIT 15=1 PROGRAM SHOULD BE POWER FAILED WHILE RUNNING
464          ;*              =0 NO POWER FAIL DESIRED
465          ;*
466          ;*      BIT 14=1 PROGRAM RUN TIME IS MEMORY SIZE DEPENDENT
467          ;*              =0 RUN TIME IS NOT MEMORY SIZE DEPENDENT
468          ;*
469          ;*      BITS 13-0 MUST BE ZERO'S
470
471          000204          $$VPC=.          ;; SAVE LOCATION COUNTER
472          000046          .-46          ;; SET LOCATION COUNTER
473 000046 027324          .WORD  $ENDAD          ;; SET LOC.46 TO ADDRESS $ENDAD
474          000052          .-52          ;; SET LOCATION COUNTER
475 000052 000000          .WORD  0          ;; SET LOC.52 TO ZERO
476          000204          . $VPC          ;; RESTORE LOCATION COUNTER
477
```

```

478      ::*****
479
480      .SBTTL COMMON TAGS
481
482      :*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
483      :*USED IN THE PROGRAM.
484
485      001100      .=1100
486
487      001100      $CMTAG:      ::START OF COMMON TAGS
488      001100      000000      $PASS:      .WORD      0      ::CONTAINS PASS COUNT
489      001102      000      $STNM:      .BYTE      0      ::CONTAINS THE TEST NUMBER
490      001103      000      $ERFLG:      .BYTE      0      ::CONTAINS ERROR FLAG
491      001104      000000      $ICNT:      .WORD      0      ::CONTAINS SUBTEST ITERATION COUNT
492      001106      000000      $LPADR:      .WORD      0      ::CONTAINS SCOPE LOOP
493      001110      000000      $LPERR:      .WORD      0      ::CONTAINS SCOPE RETURN FOR ERRORS
494      001112      000000      $ERTTL:      .WORD      0      ::CONTAINS TOTAL ERRORS DETECTED
495      001114      000      $ITEMB:      .BYTE      0      ::CONTAINS ITEM CONTROL BYTE
496      001115      001      $ERMAX:      .BYTE      1      ::CONTAINS MAX. ERRORS PER TEST
497      001116      000000      $ERRPC:      .WORD      0      ::CONTAINS PC OF LAST ERROR INSTRUCTION
498      001120      000000      $GDADR:      .WORD      0      ::CONTAINS OF 'GOOD' DATA
499      001122      000000      $BDADR:      .WORD      0      ::CONTAINS OF 'BAD' DATA
500      001124      000000      $GDDAT:      .WORD      0      ::CONTAINS 'GOOD' DATA
501      001126      000000      $BDDAT:      .WORD      0      ::CONTAINS 'BAD' DATA
502      001130      000000      000000      000000      .WORD      0,0,0      ::RESERVED--NOT TO BE USED
503      001136      177560      $TKS:      177560      ::TTY KBD STATUS
504      001140      177562      $TKB:      177562      ::TTY KBD BUFFER
505      001142      177564      $TPS:      177564      ::TTY PRINTER STATUS REG.
506      001144      177566      $TPB:      177566      ::TTY PRINTER BUFFER REG.
507      001146      000      $NULL:      .BYTE      0      ::CONTAINS NULL CHARACTER FOR FILLS
508      001147      002      $FILLS:      .BYTE      2      ::CONTAINS # OF FILLER CHARACTERS REQUIRED
509      001150      012      $FILLC:      .BYTE      12      ::INSERT FILL CHARS. AFTER A 'LINE FEED'
510      001151      000      $TPFLG:      .BYTE      0      ::'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
511      001152      000000      $REGAD:      .WORD      0      ::CONTAINS THE FROM
512      ::WHICH ($REGO) WAS OBTAINED
513      001154      000000      $REG0:      .WORD      0      ::CONTAINS (($REGAD)+0)
514      001156      000000      $REG1:      .WORD      0      ::CONTAINS (($REGAD)+2)
515      001160      000000      $REG2:      .WORD      0      ::CONTAINS (($REGAD)+4)
516      001162      000000      $REG3:      .WORD      0      ::CONTAINS (($REGAD)+6)
517      001164      000000      $REG4:      .WORD      0      ::CONTAINS (($REGAD)+10)
518      001166      000000      $REG5:      .WORD      0      ::CONTAINS (($REGAD)+12)
519      001170      000000      $REG6:      .WORD      0      ::CONTAINS (($REGAD)+14)
520      001172      000000      $REG7:      .WORD      0      ::CONTAINS (($REGAD)+16)
521      001174      000000      $REG10:      .WORD      0      ::CONTAINS (($REGAD)+20)
522      001176      000000      $REG11:      .WORD      0      ::CONTAINS (($REGAD)+22)
523      001200      000000      $REG12:      .WORD      0      ::CONTAINS (($REGAD)+24)
524      001202      000000      $REG13:      .WORD      0      ::CONTAINS (($REGAD)+26)
525      001204      000000      $REG14:      .WORD      0      ::CONTAINS (($REGAD)+30)
526      001206      000000      $REG15:      .WORD      0      ::CONTAINS (($REGAD)+32)
527      001210      000000      $REG16:      .WORD      0      ::CONTAINS (($REGAD)+34)
528      001212      000000      $REG17:      .WORD      0      ::CONTAINS (($REGAD)+36)
529      001214      000000      $REG20:      .WORD      0      ::CONTAINS (($REGAD)+40)
530      001216      000000      $REG21:      .WORD      0      ::CONTAINS (($REGAD)+42)
531      001220      000000      $REG22:      .WORD      0      ::CONTAINS (($REGAD)+44)
532      001222      000000      $REG23:      .WORD      0      ::CONTAINS (($REGAD)+46)
533      001224      000000      $TMP0:      .WORD      0      ::USER DEFINED
    
```

```

534 001226 000000 $TMP1: .WORD 0 ;;USER DEFINED
535 001230 000000 $TMP2: .WORD 0 ;;USER DEFINED
536 001232 000000 $TMP3: .WORD 0 ;;USER DEFINED
537 001234 000000 $TMP4: .WORD 0 ;;USER DEFINED
538 001236 000000 $TMP5: .WORD 0 ;;USER DEFINED
539 001240 000000 $TMP6: .WORD 0 ;;USER DEFINED
540 001242 000000 $TMP7: .WORD 0 ;;USER DEFINED
541 001244 000000 $TMP10: .WORD 0 ;;USER DEFINED
542 001246 000000 $TMP11: .WORD 0 ;;USER DEFINED
543 001250 000000 $TMP12: .WORD 0 ;;USER DEFINED
544 001252 000000 $TMP13: .WORD 0 ;;USER DEFINED
545 001254 000000 $TMP14: .WORD 0 ;;USER DEFINED
546 001256 000000 $TMP15: .WORD 0 ;;USER DEFINED
547 001260 000000 $TMP16: .WORD 0 ;;USER DEFINED
548 001262 000000 $TMP17: .WORD 0 ;;USER DEFINED
549 001264 000000 $TMP20: .WORD 0 ;;USER DEFINED
550 001266 000000 $TMP21: .WORD 0 ;;USER DEFINED
551 001270 000000 $TMP22: .WORD 0 ;;USER DEFINED
552 001272 000000 $TMP23: .WORD 0 ;;USER DEFINED
553 001274 000000 $TIMES: 0 ;;MAX. NUMBER OF ITERATIONS
554 001276 000000 $ESCAPE: 0 ;;ESCAPE ON ERROR
555 001300 177607 000377 $BELL: .ASCIZ <207><377><377> ;;CODE FOR BELL
556 001304 077 $QUES: .ASCII /?/ ;;QUESTION MARK
557 001305 015 $CRLF: .ASCII <15> ;;CARRIAGE RETURN
558 001306 000C12 $LF: .ASCIZ <12> ;;LINE FEED
559 001310 000 KB11E: .BYTE 0 ;:1174 WITHOUT MP CACHE FLAG
560 001311 000 KB11EM: .BYTE 0 ;:1174 WITH MP CACHE FLAG
561 001312 000 KB11CM: .BYTE 0 ;:KB11CM FLAG (1170 WITH MP MODS)
562 001313 000 CISP: .BYTE 0 ;:CISP OPTION PRESENT FLAG
563
564 ;OPCODE FOR MFPT INSTRUCTION (AVAILABLE ON KB11-E AND KB11-EM ONLY)
565 000007 MFPT=7
    
```

```

566      ;:*****
567
568      .SBTTL  ERROR POINTER TABLE
569
570      ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
571      ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
572      ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
573      ;*NOTE1:      IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
574      ;*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
575
576      ;*      EM      ;;POINTS TO THE ERROR MESSAGE
577      ;*      DH      ;;POINTS TO THE DATA HEADER
578      ;*      DT      ;;POINTS TO THE DATA
579      ;*      DF      ;;POINTS TO THE DATA FORMAT
580
581
582      001314      $ERRTB:
583
584
585
586      ;ERROR TABLE FOR ERROR TYPE OUT:
587      ;ITEM 1
588      001314      036504      050056      052202      .WORD      EM1,DH1,DT1,DF1
589      001322      052005
590      ;ITEM 0
591      001324      000000      000000      000000      .WORD      0,0,0,0
592      001332      000000
593      ;ITEM 0
594      001334      000000      000000      000000      .WORD      0,0,0,0
595      001342      000000
596      ;ITEM 0
597      001344      000000      000000      000000      .WORD      0,0,0,0
598      001352      000000
599      ;ITEM 0
600      001354      000000      000000      000000      .WORD      0,0,0,0
601      001362      000000
602      ;ITEM 0
603      001364      000000      000000      000000      .WORD      0,0,0,0
604      001372      000000
605      ;ITEM 0
606      001374      000000      000000      000000      .WORD      0,0,0,0
607      001402      000000
608      ;ITEM 0
609      001404      000000      000000      000000      .WORD      0,0,0,0
610      001412      000000
611      ;ITEM 0
612      001414      000000      000000      000000      .WORD      0,0,0,0
613      001422      000000
614      ;ITEM 0
615      001424      000000      000000      000000      .WORD      0,0,0,0
616      001432      000000
617      ;ITEM 0
618      001434      000000      000000      000000      .WORD      0,0,0,0
619      001442      000000
620      ;ITEM 14
621      001444      036571      050131      052214      .WORD      EM14,DH14,DT14,DF14
    
```

622	001452	052011				
623					:ITEM 15	
624	001454	036630	050224	052230	.WORD	EM15,DH15,DT15,DF15
625	001462	052016				
626					:ITEM 0	
627	001464	000000	000000	000000	.WORD	0,0,0,0
628	001472	000000				
629					:ITEM 0	
630	001474	000000	000000	000000	.WORD	0,0,0,0
631	001502	000000				
632					:ITEM 0	
633	001504	000000	000000	000000	.WORD	0,0,0,0
634	001512	000000				
635					:ITEM 0	
636	001514	000000	000000	000000	.WORD	0,0,0,0
637	001522	000000				
638					:ITEM 0	
639	001524	000000	000000	000000	.WORD	0,0,0,0
640	001532	000000				
641					:ITEM 0	
642	001534	000000	000000	000000	.WORD	0,0,0,0
643	001542	000000				
644					:ITEM 0	
645	001544	000000	000000	000000	.WORD	0,0,0,0
646	001552	000000				
647					:ITEM 0	
648	001554	000000	000000	000000	.WORD	0,0,0,0
649	001562	000000				
650					:ITEM 0	
651	001564	000000	000000	000000	.WORD	0,0,0,0
652	001572	000000				
653					:ITEM 0	
654	001574	000000	000000	000000	.WORD	0,0,0,0
655	001602	000000				
656					:ITEM 0	
657	001604	000000	000000	000000	.WORD	0,0,0,0
658	001612	000000				
659					:ITEM 0	
660					:ITEM 0	
661	001614	000000	000000	000000	.WORD	0,0,0,0
662	001622	000000				
663					:ITEM 0	
664	001624	000000	000000	000000	.WORD	0,0,0,0
665	001632	000000				
666					:ITEM 0	
667	001634	000000	000000	000000	.WORD	0,0,0,0
668	001642	000000				
669					:ITEM 0	
670	001644	000000	000000	000000	.WORD	0,0,0,0
671	001652	000000				
672					:ITEM 0	
673	001654	000000	000000	000000	.WORD	0,0,0,0
674	001662	000000				
675					:ITEM 0	
676	001664	000000	000000	000000	.WORD	0,0,0,0
677	001672	000000				



678					:ITEM 0		
679	001674	000000	000000	000000	.WORD	0,0,0,0	
680	001702	000000					
681					:ITEM 0		
682	001704	000000	000000	000000	.WORD	0,0,0,0	
683	001712	000000					
684					:ITEM 0		
685	001714	000000	000000	000000	.WORD	0,0,0,0	
686	001722	000000					
687					:ITEM 0		
688	001724	000000	000000	000000	.WORD	0,0,0,0	
689	001732	000000					
690					:ITEM 0		
691	001734	000000	000000	000000	.WORD	0,0,0,0	
692	001742	000000					
693					:ITEM 0		
694	001744	000000	000000	000000	.WORD	0,0,0,0	
695	001752	000000					
696					:ITEM 0		
697	001754	000000	000000	000000	.WORD	0,0,0,0	
698	001762	000000					
699					:ITEM 0		
700	001764	000000	000000	000000	.WORD	0,0,0,0	
701	001772	000000					
702					:ITEM 0		
703	001774	000000	000000	000000	.WORD	0,0,0,0	
704	002002	000000					
705					:ITEM 0		
706	002004	000000	000000	000000	.WORD	0,0,0,0	
707	002012	000000					
708					:ITEM 0		
709	002014	000000	000000	000000	.WORD	0,0,0,0	
710	002022	000000					
711					:ITEM 0		
712	002024	000000	000000	000000	.WORD	0,0,0,0	
713	002032	000000					
714					:ITEM 0		
715	002034	000000	000000	000000	.WORD	0,0,0,0	
716	002042	000000					
717					:ITEM 0		
718	002044	000000	000000	000000	.WORD	0,0,0,0	
719	002052	000000					
720					:ITEM 55		
721							
722	002054	036700	050250	052236	.WORD	EM55,DH55,DT55,DF55	
723	002062	052020					
724					:ITEM 56		
725	002064	037044	050250	052236	.WORD	EM56,DH56,DT56,DF56	
726	002072	052020					
727					:ITEM 57		
728	002074	037211	050250	052236	.WORD	EM57,DH57,DT57,DF57	
729	002102	052020					
730					:ITEM 60		
731	002104	037333	050250	052236	.WORD	EM60,DH60,DT60,DF60	
732	002112	052020					
733					:ITEM 61		

734	002114	037457	050250	052236	.WORD	EM61,DH61,DT61,DF61
735	002122	052020				
736					:ITEM 62	
737	002124	037607	050250	052236	.WORD	EM62,DH62,DT62,DF62
738	002132	052020				
739					:ITEM 63	
740	002134	037735	050325	052250	.WORD	EM63,DH63,DT63,DF63
741	002142	052024				
742					:ITEM 64	
743	002144	040154	050427	052262	.WORD	EM64,DH64,DT64,DF64
744	002152	052024				
745					:ITEM 65	
746	002154	040353	050502	052272	.WORD	EM65,DH65,DT65,DF65
747	002162	052024				
748					:ITEM 66	
749	002164	040736	050604	052304	JRD	EM66,DH66,DT66,DF66
750	002172	052024				
751					:ITEM 67	
752	002174	041020	050657	052262	.WORD	EM67,DH67,DT67,DF67
753	002202	052024				
754					:ITEM 70	
755	002204	041235	050657	052262	.WORD	EM70,DH70,DT70,DF70
756	002212	052024				
757					:ITEM 71	
758	002214	041513	050657	052262	.WORD	EM71,DH71,DT71,DF71
759	002222	052024				
760					:ITEM 72	
761	002224	041771	050657	052262	.WORD	EM72,DH72,DT72,DF72
762	002232	052024				
763					:ITEM 73	
764	002234	042213	050657	052262	.WORD	EM73,DH73,DT73,DF73
765	002242	052024				
766					:ITEM 74	
767	002244	042477	050657	052262	.WORD	EM74,DH74,DT74,DF74
768	002252	052024				
769					:ITEM 75	
770						
771	002254	042763	050754	052320	.WORD	EM75,DH75,DT75,DF75
772	002262	052031				
773					:ITEM 76	
774	002264	042763	050754	052334	.WORD	EM76,DH76,DT76,DF76
775	002272	052031				
776					:ITEM 77	
777	002274	043122	051051	052350	.WORD	EM77,DH77,DT77,DF77
778	002302	052036				
779					:ITEM 0	
780	002304	000000	000000	000000	.WORD	0,0,0,0
781	002312	000000				
782					:ITEM 0	
783	002314	000000	000000	000000	.WORD	0,0,0,0
784	002322	000000				
785					:ITEM 0	
786	002324	000000	000000	000000	.WORD	0,0,0,0
787	002332	000000				
788					:ITEM 0	
789	002334	000000	000000	000000	.WORD	0,0,0,0

790	002342	000000					
791					:ITEM 0		
792	002344	000000	000000	000000	.WORD	0,0,0,0	
793	002352	000000					
794					:ITEM 0		
795	002354	000000	000000	000000	.WORD	0,0,0,0	
796	002362	000000					
797					:ITEM 0		
798	002364	000000	000000	000000	.WORD	0,0,0,0	
799	002372	000000					
800					:ITEM 0		
801	002374	000000	000000	000000	.WORD	0,0,0,0	
802	002402	000000					
803					:ITEM 0		
804	002404	000000	000000	000000	.WORD	0,0,0,0	
805	002412	000000					
806					:ITEM 0		
807	002414	000000	000000	000000	.WORD	0,0,0,0	
808	002422	000000					
809					:ITEM 0		
810	002424	000000	000000	000000	.WORD	0,0,0,0	
811	002432	000000					
812							
813					:ITEM 0		
814	002434	000000	000000	000000	.WORD	0,0,0,0	
815	002442	000000					
816					:ITEM 0		
817	002444	000000	000000	000000	.WORD	0,0,0,0	
818	002452	000000					
819					:ITEM 0		
820	002454	000000	000000	000000	.WORD	0,0,0,0	
821	002462	000000					
822					:ITEM 0		
823	002464	000000	000000	000000	.WORD	0,0,0,0	
824	002472	000000					
825					:ITEM 117		
826	002474	043260	050754	052334	.WORD	EM117,DH117,DT117,DF117	
827	002502	052031					
828					:ITEM 120		
829	002504	043407	051075	052376	.WORD	EM120,DH120,DT120,DF120	
830	002512	052050					
831					:ITEM 121		
832	002514	043622	051151	052466	.WORD	EM121,DH121,DT121,DF121	
833	002522	052103					
834					:ITEM 122		
835	002524	044023	051213	052500	.WORD	EM122,DH122,DT122,DF122	
836	002532	052107					
837					:ITEM 123		
838	002534	044153	051275	052500	.WORD	EM123,DH123,DT123,DF123	
839	002542	052107					
840					:ITEM 124		
841	002544	044354	050131	052512	.WORD	EM124,DH124,DT124,DF124	
842	002552	052113					
843					:ITEM 0		
844	002554	000000	000000	000000	.WORD	0,0,0,0	
845	002562	000000					

846					:ITEM 0		
847	002564	000000	000000	000000	.WORD	0,0,0,0	
848	002572	000000					
849					:ITEM 127		
850	002574	044562	051445	052532	.WORD	EM127,DH127,DT127,DF127	
851	002602	052137					
852					:ITEM 130		
853	002604	044744	051507	052564	.WORD	EM130,DH130,DT130,DF130	
854	002612	052123					
855							
856					:ITEM 131		
857	002614	045016	051565	052576	.WORD	EM131,DH131,DT131,DF131	
858	002622	052142					
859					:ITEM 132		
860	002624	047130	051335	052532	.WORD	EM132,DH132,DT132,DF132	
861	002632	052123					
862					:ITEM 133		
863	002634	047267	051372	052542	.WORD	EM133,DH133,DT133,DF133	
864	002642	052127					
865					:ITEM 134		
866	002644	047441	051644	052624	.WORD	EM134,DH134,DT134,DF134	
867	002652	052154					
868					:ITEM 135		
869	002654	047607	051051	052644	.WORD	EM135,DH135,DT135,DF135	
870	002662	052163					
871					:ITEM 0		
872	002664	000000	000000	000000	.WORD	0,0,0,0	
873	002672	000000					
874					:ITEM 0		
875	002674	000000	000000	000000	.WORD	0,0,0,0	
876	002702	000000					
877					:ITEM 140		
878	002704	045243	047047	047116	.WORD	EM140,DH140,DT140,DF140	
879	002712	047112					
880					:ITEM 141		
881	002714	045604	047047	047116	.WORD	EM141,DH141,DT141,DF141	
882	002722	047112					
883					:ITEM 142		
884	002724	046144	047047	047116	.WORD	EM142,DH142,DT142,DF142	
885	002732	047112					
886					:ITEM 143		
887	002734	046506	047047	047116	.WORD	EM143,DH143,DT143,DF143	
888	002742	047112					
889					:ITEM 0		
890	002744	000000	000000	000000	.WORD	0,0,0,0	
891	002752	000000					
892					:ITEM 0		
893	002754	000000	000000	000000	.WORD	0,0,0,0	
894	002762	000000					
895					:ITEM 0		
896	002764	000000	000000	000000	.WORD	0,0,0,0	
897	002772	000000					
898					:ITEM 0		
899	002774	000000	000000	000000	.WORD	0,0,0,0	
900	003002	000000					
901					:ITEM 150		

902 003004 047772 051721 052672 .WORD EM150,DH150,DT150,DF150  
 903 003012 052175

904  
 905  
 906 003014 005037 001102  
 907 003020 012737 000340 177776  
 908 003026 012706 001100  
 909 003032 005026  
 910 003034 022706 001136  
 911 003040 001374  
 912 003042 012706 001100  
 913 003046 012737 027360 000020  
 914 003054 012737 000340 000022  
 915 003062 012737 027642 000030  
 916 003070 012737 000340 000032  
 917 003076 012737 031014 000034  
 918 003104 012737 000340 000036  
 919 003112 012737 031074 000024  
 920 003120 012737 000340 000026  
 921 003126 013737 027254 027246  
 922 003134 005037 001274  
 923 003140 005037 001276  
 924 003144 112737 000001 001115  
 925 003152 012737 003152 001106  
 926 003160 012737 003160 001110  
 927 003166 005227 177777  
 928 003172 001040  
 929 003174 022737 027324 000042  
 930 003202 001434  
 931 003204 104400 003212  
 932 003210 000431

```

START: CLR $STNM
MOV #340,@MPS ;;LOCK OUT ALL INTERRUPTS
MOV #SCMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
CLR (R6)+ ;;CLEAR MEMORY LOCATION
CMP #STKS,R6 ;;DONE?
BNE .-6 ;;LOOP BACK IF NO
MOV #STACK,SP ;;SETUP THE STACK POINTER
MOV #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
MOV #340,@IOTVEC+2 ;;LEVEL 7
MOV #ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
MOV #340,@EMTVEC+2 ;;LEVEL 7
MOV #STRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
MOV #340,@TRAPVEC+2;LEVEL 7
MOV #SPWRDN,@PWRVEC ;;POWER FAILURE VECTOR
MOV #340,@PWRVEC+2 ;;LEVEL 7
MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
MOVB #1,$ERMAX ;;ALLOW ONE ERROR PER TEST
MOV #,$SLPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
MOV #,$SLPERR ;;SETUP THE ERROR LOOP ADDRESS
INC #-1 ;;FIRST TIME?
BNE 64$ ;;BRANCH IF NO
CMP #SENDAD,@#42 ;;ACT-11?
BEQ 64$ ;;BRANCH IF YES
TYPE ,65$ ;;TYPE ASCII STRING
BR 64$ ;;GET OVER THE ASCIIZ
;;65$: .ASCIIZ <CRLF>'CEKBC-C PDP 11/70-74MP CACHE DIAGNOSTIC PART 1'<CRLF>
64$:
    
```

933  
 934 003274  
 935  
 936  
 937  
 938  
 939  
 940  
 941  
 942  
 943  
 944  
 945  
 946  
 947  
 948  
 949  
 950  
 951  
 952  
 953  
 954  
 955  
 956  
 957 003274 105037 001312

: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.

```

*** TEST FOR VARIOUS KB11 PROCESSORS ***
*THIS ROUTINE POLES THE RESULTS OF ATTEMPTS TO SET TO ONE
*CERTAIN CRITICAL BITS THAT ARE KNOWN TO BE OPERATIVE ON A KB11CM,
*OR KB11EM PROCESSOR. IF TWO OUT OF FOUR OF THE TESTS ARE
*POSITIVE THEN THE KB11CM OR KB11EM FLAG IS SET,IF LESS THAN TWO OF THE
*TESTS ARE POSITIVE THEN THE KB11E FLAG OR NO FLAG IS SET. THE DETERMINATION
*OF WHICH PAIR IS VALID IS BASED ON THE RESULTS OF EXECUTING AN MFPT OPCODE
*(OPCODE 7). IF THIS INSTRUCTION TRAPS THIS IS AN KB11CM OR
*A PLAIN 1170 (KB11-B OR KB11-C). IF THE INSTRUCTION DOES NOT TRAP THEN
*THIS IS A KB11-E OR KB11-EM.
    
```

```

KBTST: CLRB @KB11CM ;RESET THE MP FLAG
    
```

958	003300	005037	001310		CLR	@#KB11E	;CLEAR KB11E AND KB11EM FLAGS
959	003304	012737	003542	000010	MOV	#MFPTTR,@#RESVEC	;SET UP TRAP ADDRESS FOR MFPT AT RESERV VECTOR
960	003312	000007			MFPT		;EXECUTE MFPT. WILL TRAP ON 1170 (KB11B/C) OR
961							;KB11CM
962	003314	012737	000001	001310	MOV	#1,@#KB11E	;HERE IF KB11E OR KB11EM. SET FLAG
963	003322	005037	177750		T1: CLR	@#MAINT	;CLEAR THE MAINTENANCE REGISTER
964	003326	005005			CLR	R5	;RESET THE TEST COUNTER
965	003330	012700	177746		MOV	#CONTRL,R0	;GET THE ADDRESS OF...
966	003334	012701	177750		MOV	#MAINT,R1	;CCR,MAINT,AND MAPH00...
967	003340	012702	170202		MOV	#MAPH00,R2	;AND PLACE IN R0-R2
968	003344	052710	040000		BIS	#BIT14,(R0)	;TRY TO SET IVSS BIT
969	003350	032710	040000		BIT	#BIT14,(R0)	;DID IT SET?
970	003354	001403			BEQ	T2	;NO,GO TO NEXT TEST
971	003356	042710	040000		BIC	#BIT14,(R0)	;CLEAR IT.
972	003362	005205			INC	R5	;TEST IS POSITIVE
973	003364	052711	000001		T2: BIS	#BIT0,(R1)	;SET EDMA IN MAINT REGISTER
974	003370	032711	000001		BIT	#BIT0,(R1)	
975	003374	001410			BEQ	T3	
976	003376	052710	004000		BIS	#BIT11,(R0)	;TRY TO SET DMMA IN CCR
977	003402	032710	004000		BIT	#BIT11,(R0)	
978	003406	001403			BEQ	T3	
979	003410	042710	004000		BIC	#BIT11,(R0)	
980	003414	005205			INC	R5	
981	003416	042711	000001		T3: BIC	#BIT0,(R1)	;MAKE SURE EDMA IS CLEAR
982	003422	052737	100000	172300	BIS	#BIT15,KIPDR0	;TRY TO SET BYP ON A PDR
983	003430	032737	100000	172300	BIT	#BIT15,KIPDR0	
984	003436	001404			BEQ	T4	
985	003440	042737	100000	172300	BIC	#BIT15,KIPDR0	
986	003446	005205			INC	R5	
987	003450	052712	100000		T4: BIS	#BIT15,(R2)	;TRY TO SET BYP ON UNIBUS MAP
988	003454	032712	100000		BIT	#BIT15,(R2)	
989	003460	001403			BEQ	T.END	
990	003462	042712	100000		BIC	#BIT15,(R2)	
991	003466	005205			INC	R5	
992	003470	022705	000002		T.END: CMP	#2,R5	;IS THE RESULT OF THE TEST >=2
993	003474	101021			BHI	2\$	;NO,THIS IS A KB11E OR KB11-B/C (11/70)
994	003476	005000			CLR	R0	
995	003500	005037	177746		CLR	@#CONTRL	
996	003504	013701	177746		3\$: MOV	@#CONTRL,R1	
997	003510	001402			BEQ	4\$	
998	003512	005200			INC	R0	
999	003514	001373			BNE	3\$	
1000	003516				4\$:		
1001	003516	005737	001310		TST	@#KB11E	;IS IS A KB11-E OR KB11-EM?
1002	003522	001404			BEQ	1\$	;BR IF NEITHER. MUST BE KB11CM
1003	003524	012737	000400	001310	MOV	#BIT8,@#KB11E	;SET UPPER BYTE (KB11-EM)
1004	003532	000402			BR	2\$	;DONE
1005	003534	105237	001312		1\$: INCB	@#KB11CM	;YES, FLAG THIS AS A MODIFIED PROCESSOR
1006	003540	000403			2\$: BR	ENDKB	;DONE DETERMINING WHICH CPU
1007							
1008	003542				MFPTTR:		
1009	003542	012716	003322		MOV	#T1,(SP)	;HERE IF MFPT TRAPPED. SEE IF 1170 OR KB11CM
1010	003546	000002			RTI		;SET UP RETURN ADDRESS FOR RTI
1011	003550				ENDKB:		;RETURN
1012	003550	005227	177777		INC	#-1	;FIRST TIME?
1013	003554	001026			BNE	100\$	;BR IF NO

```

1014 003556 104400 036361 TYPE ,MSG1 ;<15><12>CPU UNDER TEST FOUND TO BE A
1015 003562 005737 001310 TST @#KB11E ;IS THIS A KB11-E OR KB11-EM?
1016 003566 001011 BNE 101$ ;BR IF EITHER ONE
1017 003570 105737 001312 TSTB @#KB11CM ;IS IT A KB11CM
1018 003574 001003 BNE 1$ ;BR IF IT IS
1019 003576 104400 036431 TYPE ,MSG3 ;KB11-B/C<15><12>
1020 003602 000413 BR 100$ ;SKIP OTHER MESSAGE
1021 003604 104400 036443 1$: TYPE ,MSG4 ;KB-CM11<15><12>
1022 003610 000410 BR 100$ ;SKIP CISP MESSAGE
1023 003612 105737 001310 101$: TSTB @#KB11E ;IS IT A KB11-E?
1024 003616 001403 BEQ 102$ ;BR IF NOT. MUST BE KB11-EM
1025 003620 104400 036474 TYPE ,MSG5 ;KB11-E<15><12>
1026 003624 000402 BR 100$ ;SKIP KB11-EM MESSAGE
1027 003626 104400 036420 102$: TYPE ,MSG2 ;KB11-EM<15><12>
1028 003632 100$:
1029 *****
1030 ;SIZE MEMORY AND COMPARE IT WITH THE SYSTEM SIZE REGISTER
1031 ;PRINT A WARNING IF THEY DISAGREE.
1032 003632 052737 000200 031276 BIS #BIT07,$KT11
1033 003640 004737 031230 JSR PC,$SIZE
1034 003644 062737 000037 031614 ADD #37,$LSTBK ;ADJUST THE SIZE FOR PROPER
1035 ;COMPARISON TO SIZE REGISTER
1036 003652 023737 177760 031614 CMP @#SIZELO,$LSTBK ;SIZE REGISTER EQUAL TO ACTUAL SIZE?
1037 003660 001546 BEQ OKSIZ
1038 003662 104400 003670 TYPE ,65$ ;:TYPE ASCIZ STRING
1039 003666 000433 BR 64$ ;:GET OVER THE ASCIZ
1040 ;:65$: .ASCIZ <15><12>/WARNING- THE SIZE OF MEMORY IS DIFFERENT FROM THAT/
1041 64$:
1042 003756 104400 003764 TYPE ,67$ ;:TYPE ASCIZ STRING
1043 003762 000425 BR 66$ ;:GET OVER THE ASCIZ
1044 ;:67$: .ASCIZ <15><12>/INDICATED BY THE SYSTEM SIZE REGISTER./
1045 66$:
1046 004036 104400 004044 TYPE ,69$ ;:TYPE ASCIZ STRING
1047 004042 000421 BR 68$ ;:GET OVER THE ASCIZ
1048 ;:69$: .ASCIZ <15><12>/ SIZEHI SIZELO ACTUAL/
1049 68$:
1050 004106 104400 001305 TYPE ,$CRLF
1051 004112 013746 177762 MOV @#SIZEHI,-(SP) ;:SAVE @#SIZEHI FOR TYPEOUT
1052 004116 104404 TYPOS ;:GO TYPE--OCTAL ASCII
1053 004120 006 .BYTE 6 ;:TYPE 6 DIGIT(S)
1054 004121 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
1055 004122 104400 004130 TYPE ,71$ ;:TYPE ASCIZ STRING
1056 004126 000404 BR 70$ ;:GET OVER THE ASCIZ
1057 ;:71$: .ASCIZ / /
1058 70$:
1059 004140 013746 177760 MOV @#SIZELO,-(SP) ;:SAVE @#SIZELO FOR TYPEOUT
1060 004144 104404 TYPOS ;:GO TYPE--OCTAL ASCII
1061 004146 006 .BYTE 6 ;:TYPE 6 DIGIT(S)
1062 004147 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
1063 004150 104400 004156 TYPE ,73$ ;:TYPE ASCIZ STRING
1064 004154 000404 BR 72$ ;:GET OVER THE ASCIZ
1065 ;:73$: .ASCIZ / /
1066 72$:
1067 004160 013746 031614 MOV $LSTBK,-(SP) ;:SAVE $LSTBK FOR TYPEOUT
1068 004172 104404 TYPOS ;:GO TYPE--OCTAL ASCII
1069 004174 006 .BYTE 6 ;:TYPE 6 DIGIT(S)
    
```

```

1070 004175 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
1071 004176 OKSIZ:
1072
1073 ;:*****
1074 004176 005237 032526 LOOP: INC MONF ;:INCREMENT THE FLAG WHICH INDICATES
1075 004202 001013 BNE TOP ;:WHETHER OR NOT THE TOP OF MEMORY
1076 ;:IN THE FIRST 28K HAS BEEN SAVED.
1077 004204 013737 000060 032524 MOV @TKVEC,MONTTY ;:SAVE THE INITIAL CONTENTS OF THE TTY KEYBOARD
1078 ;:VECTOR.
1079 004212 012700 002734 MOV #^D1500,R0 ;:IF NOT THEN SAVE IT.
1080 004216 012701 052710 MOV #BOTTOM+4,R1 ;:SAVE IT AT THE BOTTOM OF THIS PROGRAM.
1081 004222 012702 160000 MOV #160000,R2 ;:GET THE ADDRESS OF THE END OF THE MONITOR.
1082 004226 014221 1$: MOV -(R2),(R1)+ ;:SAVE 1500 (DEC) LOCATIONS (WORDS)
1083 004230 077002 SOB R0,1$
1084 004232 012737 000044 177770 TOP: MOV #44,@177770 ;:SET TO SYNC SCOPE (OSCILLOSCOPE)
1085 ;:ON A NOP INSTRUCTION.
1086
1087 004240 012737 032372 000060 MOV #RESMON,@TKVEC ;:SET UP THE KEYBORD INTERRUPT VECTOR.
1088 004246 012737 000340 000062 MOV #340,@TKVEC+2
1089 004254 005077 174660 CLR @STKB ;:MAKE SURE THE BUFFER IS CLEAR
1090 004260 152777 000100 174650 BISB #BIT6,@STKS ;:TURN ON INTERRUPT ENABLE FOR THE KEYBOARD.
1091
1092 004266 012737 031736 000004 MOV #CPSPUR,@ERRVEC ;:SET UP FOR UNEXPECTED ERRORS.
1093 004274 012737 031764 000114 MOV #SPUR,@CACHVEC
1094
1095
1096 ;:*****
1097 ;:*TEST 1 CACHE REGISTERS RESPONSE TEST
1098 ;:*
1099 ;:*REFERENCE EACH CACHE REGISTER MAKING SURE SUCH
1100 ;:*REFERENCES DO NOT TIME OUT.
1101 ;:*
1102 ;:*****
1103 004302 000004 TST1: SCOPE
1104 004304 012737 000040 001274 MOV #40,$TIMES ;:DO 40 ITERATIONS
1105 000001 JA $TN-1
1106 ;:SET THE SKAD REGISTER
1107 004312 012737 004656 032110 MOV #TST2,SKAD ;:IN CASE THE TEST ABORTS.
1108
1109 004320 113737 001102 001224 MOVB $TSTNM,$TMP0
1110 004326 012737 031764 000114 MOV #SPUR,@CACHVEC ;:EXPECT NO PARITY ERRORS.
1111 004334 012701 032320 MOV #LOAFLG,R1 ;:CLEAR THE REGISTER FLAGS
1112 004340 012700 000014 MOV #14,R0
1113 004344 005021 64$: CLR (R1)+
1114 004346 077002 SOB R0,64$
1115 004350 013737 000004 004426 MOV @ERRVEC,JATMP ;:SAVE THE OLD CONTENTS OF VECTOR ERRVEC.
1116 004356 012737 004430 000004 MOV #JAERR,@ERRVEC ;:SET UP THE TIME OUT
1117 ;:VECTOR
1118 004364 012700 177740 MOV #LOADRS,R0
1119 004370 012737 004376 001110 MOV #JA1,$LPERR
1120
1121 004376 000240 JA1: NOP ;:FOR SCOPING WITH AN OSCILLOSCOPE!
1122 004400 005710 TST (R0) ;:REFERENCE EACH CACHE REGISTER
1123 ;:MAKING SURE EACH DOESN'T TIME OUT.
1124
1125 004402 062700 000002 JA2: ADD #2,R0
    
```



```

1126 004406 020027 177752      CMP      R0,#HITMIS
1127 004412 101771      BLOS     JA1
1128
1129 004414 013737 004426 000004 JA3:     MOV      JATMP,@ERRVEC ;RESET THE CPU TRAP VECTOR.
1130 004422 000137 004652      JMP      JADONE
1131
1132 004426 000000      JATMP:   .WORD    0 ;SAVE THE OLD CONTENTS OF
1133                                     ;VECTOR ERRVEC HERE.
1134
1135 004430 032737 000020 177766 JAERR:   BIT      #20,@CPUERR
1136 004436 001005      BNE     JAERR1 ;MAKE SURE THE ERROR
1137 004440 013737 004426 000004 JAERR0:  MOV      JATMP,@ERRVEC ;IF NOT RESET VECTOR ERRVEC AND GO TO
1138 004446 000177 173332      JMP      @ERRVEC ;THE ROUTINE WHICH HANDLES CPU ERRORS.
1139 004452 021627 004402      JAERR1:  CMP      (SP),#JA2 ;OTHERWISE REPORT THE FACT THAT A CACHE
1140 004456 001370      BNE     JAERR0 ;REGISTER REFERENCE TIMED OUT!
1141 004460 012637 001226      MOV      (SP)+,STMP1
1142 004464 005726      TST     (SP)+
1143 004466 010037 001232      MOV      R0,STMP3
1144 004472 012737 000077 001234      MOV      #77,STMP4
1145 004500 020027 177740      CMP      R0,#LOADRS
1146 004504 001005      BNE     JAERR2
1147 004506 012737 177777 032320      MOV      #-1,LOAFLG
1148 004514 104055      1$:     ERROR   55 ;CACHE REGISTER RESPONSE TEST FAILED
1149 004516 000451      BR      JAERR9
1150
1151 004520 020027 177742      JAERR2:  CMP      R0,#HIADRS
1152 004524 001005      BNE     JAERR3
1153 004526 012737 177777 032322      MOV      #-1,HIAFLG
1154 004534 104056      1$:     ERROR   56 ;CACHE REGISTER RESPONSE TEST FAILED
1155 004536 000441      BR      JAERR9
1156
1157 004540 020027 177744      JAERR3:  CMP      R0,#MEMERR
1158 004544 001005      BNE     JAERR4
1159 004546 012737 177777 032324      MOV      #-1,MMRFLG
1160 004554 104057      1$:     ERROR   57 ;CACHE REGISTER RESPONSE TEST FAILED
1161 004556 000431      BR      JAERR9
1162
1163 004560 020027 177746      JAERR4:  CMP      R0,#CONTRL
1164 004564 001005      BNE     JAERR5
1165 004566 012737 177777 032326      MOV      #-1,CONFLG
1166 004574 104060      1$:     ERROR   60 ;CACHE REGISTER RESPONSE TEST FAILED
1167 004576 000421      BR      JAERR9
1168
1169 004600 020027 177750      JAERR5:  CMP      R0,#MAINT
1170 004604 001005      BNE     JAERR6
1171 004606 012737 177777 032330      MOV      #-1,MANFLG
1172 004614 104061      1$:     ERROR   61 ;CACHE REGISTER RESPONSE TEST FAILED
1173 004616 000411      BR      JAERR9
1174
1175 004620 020027 177752      JAERR6:  CMP      R0,#HITMIS
1176 004624 001005      BNE     JAERR7
1177 004626 012737 177777 032332      MOV      #-1,HIMFLG
1178 004634 104062      1$:     ERROR   62 ;CACHE REGISTER RESPONSE TEST FAILED
1179 004636 000401      BR      JAERR9
1180
1181 004640 000000      JAERR7:  HALT ;???
    
```

```

1182
1183 004642 005037 177766 JAERR9: CLR @#CPUERR
1184 004646 000137 004402 JMP JA2
1185
1186 004652 005037 177766 JADONE: CLR @#CPUERR ;DONE:
1187
1188 .....
1189 :*TEST 2 CACHE REGISTERS DATA PATH, READ ZEROES TEST
1190 :*
1191 :*THIS TEST CHECKS THE ABILITY OF THE CACHE REGISTER
1192 :*DATA PATHS TO PASS 0'S BY FIRST WRITING THEN READING
1193 :*0'S AT THE CONTROL AND MAINTENANCE REGISTERS.
1194 :*
1195 .....
1196 004656 000004 TST2: SCOPE
1197 000002 JB $TN-1
1198
1199 004660 012737 005020 032110 MOV #TST3,SKAD ;SET THE SKAD REGISTER
1200 ;IN CASE THE TEST ABORTS.
1201 004666 113737 001102 001224 MOVB $STNM,$TMP0
1202 004674 012737 031764 000114 MOV #SPUR,@#CACHVEC
1203 004702 005001 CLR R1 ;;INITIALIZE
1204
1205 004704 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1206 004706 104434 SKPBMN ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
1207 004710 012737 004716 001110 MOV #JB1,$LPERR
1208 004716 005037 177746 JB1: CLR @#CONTRL ;WRITE ZEROES
1209 004722 000240 NOP ;FOR SCOPING WITH AN OSCILLOSCOPE.
1210 004724 013700 177746 1$: MOV @#CONTRL,R0 ;READ,ZEROES
1211 004730 005700 R0
1212 004732 001432 BEQ JBDONE
1213 004734 005201 INC R1
1214 004736 001372 BNE 1$ ;;ON A PDP 11/ 74 WAIT
1215 ;;FOR THE VCIP BIT IN CACHE CONT.
1216 ;;REG TO CLEAR, IN CASE A FLUSH
1217 ;;WAS INITIATED BY CLEARING VSIU BIT
1218 ;;IN CACHE CONT. REG (ABOVE)
1218 004740 005037 177750 JB2: CLR @#MAINT
1219 004744 013701 177750 MOV @#MAINT,R1
1220 004750 005701 TST R1
1221 004752 001414 BEQ JBERR2
1222
1223 004754 JBERR1: ;BOTH READ ZEROES FAILED.
1224 004754 010037 001230 MOV R0,$TMP2
1225 004760 010137 001232 MOV R1,$TMP3
1226 004764 104063 1$: ERROR 63
1227 004766 012737 177777 032326 MOV #-1,CONFLG ;SIGNAL BAD REGISTERS
1228 004774 012737 177777 032330 MOV #-1,MANFLG
1229 005002 000406 BR JBDONE
1230
1231 005004 JBERR2: ;ONLY THE READ OF THE
1232 005004 010037 001230 MOV R0,$TMP2 ;CONTROL REGISTER FAILED.
1233 005010 104064 1$: ERROR 64
1234 005012 012737 177777 032326 MOV #-1,CONFLG
1235
1236 005020 JBDONE: ;DONE:
1237
    
```

```

1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251 005020 000004
1252 005022 012737 000040 001274
1253 000003
1254
1255 005030 012737 005162 032110
1256
1257 005036 113737 001102 001224
1258
1259
1260 005044 104426
1261 005046 104430
1262 005050 012737 177777 177744
1263 005056 012737 005064 001110
1264
1265 005064 000240
1266 005066 013700 177740
1267 005072 013701 177742
1268 005076 022700 177740
1269 005102 001003
1270 005104 022701 000003
1271 005110 001424
1272
1273 005112 012737 005130 001226
1274 005120 010037 001230
1275 005124 010137 001232
1276 005130 104065
1277 005132 022700 000003
1278 005136 001403
1279 005140 012737 177777 032320
1280 005146 022700 177740
1281 005152 001403
1282 005154 012737 177777 032322
1283
1284 005162
1285
1286
1287
1288
1289
1290
1291
1292
1293
    ;*****
    ;*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.
    ;*
    ;*****
    TST3:  SCOPE
           MOV      #40,$TIMES      ;;DO 40 ITERATIONS
    JC-$TN-1
           MOV      #TST4,SKAD      ;SET THE SKAD REGISTER
           ;IN CASE THE TEST ABORTS.
           MOVB    $TSTNM,$TMP0
           SKPBAD      ;IF THE ERROR ADDRESS REG IS BAD SKIP THIS TEST.
           SKPBER     ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
           MOV      #-1,@MEMERR     ;MAKE SURE THE ERROR REGISTERS ARE UNLOCKED
           MOV      #JC1,$LPERR
    JC1:   NOP
           MOV      @LOADRS,R0      ;FOR SCOPING WITH AN OSCILLOSCOPE.
           MOV      @HIADRS,R1
           CMP      #177740,R0      ;READ THE REGISTERS.
           BNE     JCERR1
    JC2:   CMP      #3,R1
           BEQ     JCDONE
    JCERR1: MOV      #1,$TMP1      ;BAD DATA WAS READ FROM THEM..
            MOV      R0,$TMP2
            MOV      R1,$TMP3
    1$:   ERROR    65
            CMP      #3,R0
            BEQ     2$
            MOV      #-1,LOAFLG
    2$:   CMP      #177740,R0
            BEQ     JCDONE
            MOV      #-1,HIAFLG
    JCDONE:
            ;DONE:
    ;*****

```

```

;*****
;*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.

```

```

1294
1295
1296 005162 000004
1297 005164 012737 000004 001274
1298
1299          000004
1300
1301 005172 012737 005362 032110
1302
1303 005200 113737 001102 001224
1304
1305
1306 005206 104432
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316 005210 012700 177746
1317 005214 005010
1318 005216 012702 000077
1319 005222 010210
1320 005224 011001
1321 005226 042701 177700
1322 005232 020201
1323 005234 001040
1324 005236 077207
1325 005240 005010
1326 005242 105737 001311
1327 005246 001003
1328 005250 105737 001312
1329 005254 001442
1330 005256 012702 001000
1331 005262 010210
1332 005264 011001
1333 005266 001423
1334 005270 052737 000001 177750
1335 005276 072227 000002
1336 005302 010210
1337 005304 011001
1338 005306 001413
1339 005310 072227 000002
1340 005314 010210
1341 005316 011001
1342 005320 001406
1343 005322 006302
1344 005324 010210
1345 005326 011001
1346 005330 001402
1347 005332 005010
1348 005334 000412
1349

;*****
;TST4: SCOPE
;MOV #4,$TIMES ;:DO 4 ITERATIONS
;
;JD-$TN-1
;MOV #TST5,SKAD ;:SET THE SKAD REGISTER
; ;:IN CASE THE TEST ABORTS.
;MOVB $TSTNM,$TMPO
;
;SKPBCN ;:IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
;*****
;*TEST 4 CACHE CONTROL REGISTER PATTERN TEST
;*THIS TEST RUNS A COUNT PATTERN THROUGH THE LOWER 6 BITS OF THE CACHE CONTROL REGISTER
;*FOR THE PURPOSE OF CHECKING OUT THE DATA RELIABILITY OF THE REGISTER.
;*IF THE PROCESSOR HAS BEEN MODIFIED FOR MULTI PROCESSOR OPERATION THE BITS BETWEEN
;*15 AND 9, THAT ARE READ/WRITE, ARE TESTED ON AN INDIVIDUAL BASIS (KB11-EM AND
;*11/74 ).
;*****
;MOV #CONTRL,R0 ;:ADDRESS OF CONTRL TO R0
;CLR (R0) ;:CLEAR CLR
;MOV #77,R2 ;:INITIALIZE TEST PATTERN
SBT1: MOV R2,(R0) ;:WRITE IT
;MOV (R0),R1 ;:READ IT BACK
;BIC #177700,R1 ;:IGNORE <15:6>
;CMP R2,R1 ;:ARE THEY THE SAME?
;BNE JDERR1 ;:NO
SBT1.2: SOB R2,SBT1 ;:YES, ITERATE
;CLR (R0) ;:DONE WITH SUBTEST
;TSTB KB11EM ;:IS THIS A KB11-EM PROCESSOR?
;BNE ST2 ;:BR IF YES
;TSTB KB11CM ;:IS THIS A MODIFIED PROCESSOR (KB11CM)?
;BEQ JDDONE ;:NO, GO TO END OF TEST.
ST2: MOV #BIT9,R2 ;:MARCH A BIT ACROSS THE REMAINING FIELDS
;MOV R2,(R0) ;:WRITE
;MOV (R0),R1 ;:READ BACK
;BEQ JDERR1 ;:ERROR
;BIS #BIT0,@#MAINT ;:ALLOW THE DMMMA BIT (CCR<11>) TO BE SET
;ASH #2,R2 ;:SHIFT LEFT TWO
;MOV R2,(R0) ;:WRITE DMMMA
;MOV (R0),R1 ;:READ BACK
;BEQ JDERR1 ;:BAD.
;ASH #2,R2 ;:SET UP TO TEST...
;MOV R2,(R0) ;:VSIU
;MOV (R0),R1
;BEQ JDERR1
;ASL R2 ;:NOW TEST...
;MOV R2,(R0) ;:IVSS
;MOV (R0),R1
;BEQ JDERR1 ;:ERROR
;CLR (R0) ;:DONE WITH TEST
;BR JDDONE
    
```

```

1350 005336 010237 001230 JDERR1: MOV R2,$TMP2 ;REPORT THE ERROR
1351 005342 010137 001232 MOV R1,$TMP3
1352 005346 010237 001234 MOV R2,$TMP4
1353 005352 104066 ERROR 66
1354 005354 012737 177777 032326 MOV #-1,$CONFLG
1355 005362 JDDONE:
1356
1357
1358 ;*****
1359 ;*TEST 5 CACHE HIT/MISS AND CONTROL REGISTER SIMPLE MISSES TEST
1360 ;*
1361 ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
1362 ;*CONTRL REGISTER'S ABILITY TO FORCE MISSES. ZEROES ARE
1363 ;*FLOATED THROUGH THE HIT/MISS REGISTER.
1364 ;*
1365 ;*****
1366 005362 000004 TST5: SCOPE
1367 005364 012737 000040 001274 MOV #40,$TIMES ;;DO 40 ITERATIONS
1368 000005 KB=$TN-1
1369
1370 005372 012737 005714 032110 MOV #TST6,$KAD ;SET THE SKAD REGISTER
1371 ;IN CASE THE TEST ABORTS.
1372 005400 113737 001102 001224 MOVB $TSTNM,$TMP0
1373
1374
1375 005406 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1376 005410 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1377 005412 005037 005604 CLR KBFLG
1378 005416 012737 000014 177746 KB1: MOV #MOM1,@#CONTRL ;FORCE MISSES TO BOTH GROUPS.
1379 005424 012737 005416 001110 MOV #KB1,$LPERR
1380
1381 005432 012700 005442 MOV #KB2,$R0
1382 005436 012701 000020 MOV #20,$R1
1383 005442 005720 KB2: TST (R0)+
1384 005444 077102 SOB R1,KB2
1385 005446 000240 NOP ;GET SIX FORCED MISSES.
1386 005450 000240 NOP
1387 005452 000240 NOP
1388 005454 000240 NOP
1389 005456 013702 177752 MOV @#HITMIS,$R2 ;SHOULD HAVE REGISTERED
1390 005462 001051 BNE KBERR1 ;SIX MISSES.
1391
1392 005464 012737 005464 001110 KB3: MOV #KB3,$LPERR
1393 005472 012737 000054 177746 MOV #S1MOM1,@#CONTRL ;SELECT GROUP ONE, MISS GROUP
1394 005500 012700 005510 MOV #KB4,$R0 ;ZERO AND GROUP ONE.
1395 005504 012701 000020 MOV #20,$R1
1396 005510 005720 KB4: TST (R0)+
1397 005512 077102 SOB R1,KB4
1398 005514 000240 NOP
1399 005516 000240 NOP
1400 005520 000240 NOP
1401 005522 000240 NOP
1402 005524 013702 177752 MOV @#HITMIS,$R2 ;SHOULD HAVE SIX MISSES.
1403 005530 001035 BNE KBERR2
1404
1405 005532 012737 005532 001110 KB5: MOV #KB5,$LPERR
    
```

```

1406 005540 012737 000034 177746      MOV      #SOMOM1,@#CONTRL      ;SELECT GROUP 0, MISS GROUP 0
1407 005546 012700 005556              MOV      #KB6,R0              ;AND GROUP 1.
1408 005552 012701 000020              MOV      #20,R1
1409 005556 005720      KB6:    TST      (R0)+
1410 005560 077102              SOB      R1,KB6
1411 005562 000240              NOP
1412 005564 000240              NOP
1413 005566 000240              NOP
1414 005570 000240              NOP
1415 005572 013702 177752      MOV      @#HITMIS,R2          ;SHOULD HAVE SIX MISSES.
1416 005576 001021              BNE      KBERR3
1417 005600 000137 005656      JMP      KBDONE
1418
1419
1420 005604 000000      KBFLG:  .WORD 0                ;ERROR FLAG.
1421
1422 005606              KBERR1:                          ;GOT HITS WHILE FORCING
1423 005606 010237 001230      MOV      R2,$TMP2            ;MISSES TO BOTH GROUPS.
1424 005612 104072      1$:    ERROR 72
1425 005614 052737 000001 005604      BIS      #BIT0,KBFLG
1426 005622 000720              BR      KB3
1427 005624              KBERR2:                          ;GO HITS WHILE FORCING
1428 005624 010237 001230      MOV      R2,$TMP2            ;MISSES TO BOTH GROUPS
1429 005630 104073      1$:    ERROR 73                ;AND SELECTING GROUP 1
1430 005632 052737 000002 005604      BIS      #BIT1,KBFLG
1431 005640 000734              BR      KB5
1432 005642              KBERR3:                          ;GO HITS WHILE FORCING
1433 005642 010237 001230      MOV      R2,$TMP2            ;MISSES TO BOTH GROUPS
1434 005646 104074      1$:    FRROR 74                ;AND SELECTING GROUP 0.
1435 005650 052737 000004 005604      BIS      #BIT2,KBFLG
1436
1437 005656 005037 177746      KBDONE: CLR      @#CONTRL
1438 005662 022737 000007 005604      CMP      #7,KBFLG            ;IF THE TEST DETECTED
1439 005670 001003              BNE      KBD2                ;HITS FOR ALL OF THE
1440 005672 012737 177777 032346      MOV      #-1,HIMFL2          ;THREE CONDITION USED IN
1441                                          ;THE CONTROL REGISTER
1442                                          ;SIGNAL A BAD HIT/MISS
1443                                          ;REGISTER.
1444 005700 005737 005604      KBD2:  TST      KBFLG          ;IF LESS THEN THREE (BUT
1445 005704 001403              BEQ      KBD3                ;MORE THAN ZERO) CONTRL
1446 005706 012737 177777 032342      MOV      #-1,CONFL2          ;PATTERNS FAILED SIGNAL
1447                                          ;A BAD CONTROL REGISTER.
1448 005714      KBD3:                          ;DONE!
1449
1450      ;*****
1451      ;*TEST 6          CACHE HIT/MISS AND CONTROL REGISTER SIMPLE HIT TEST
1452      ;*
1453      ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
1454      ;*THE FORCE MISS BITS OF THE CONTROL REGISTER.
1455      ;*WHAT IS DONE IS TO SEE IF ANY HITS AT ALL ARE
1456      ;*POSSIBLE WITH THE CONTROL REGISTER CLEARED. THEN THE
1457      ;*SAME IS DONE WITH EACH GROUP DISABLE ONE AT A TIME.
1458      ;*BY DISABLED IS MEANT THAT THE FORCE MISS BIT IS SET
1459      ;*IN THE CONTROL REGISTER FOR THE DISABLED GROUP AND THE
1460      ;*FORCE SELECT BIT IS SET FOR THE OTHER GROUP.
1461      ;*

```

```

1462
1463 005714 000004
1464 005716 012737 000040 001274
1465 000006
1466
1467 005724 012737 006264 032110
1468
1469 005732 113737 001102 001224
1470
1471
1472 005740 104432
1473 005742 104436
1474 005744 005037 006150
1475 005750 005037 177746
1476 005754 012737 005750 001110
1477 005762 012700 005772
1478 005766 012701 000020
1479
1480 005772 005720
1481 005774 077102
1482 005776 000240
1483 006000 000240
1484 006002 000240
1485 006004 000240
1486 006006 013702 177752
1487 006012 022702 000077
1488 006016 001055
1489
1490 006020 012737 006020 001110
1491 006026 012737 000044 177746
1492 006034 012700 006044
1493 006040 012701 000020
1494 006044 005720
1495 006046 077102
1496 006050 000240
1497 006052 000240
1498 006054 000240
1499 006056 000240
1500 006060 013702 177752
1501 006064 022702 000077
1502 006070 001037
1503 006072 012737 006072 001110
1504 006100 012737 000030 177746
1505 006106 012700 006116
1506 006112 012701 000020
1507 006116 005720
1508 006120 077102
1509 006122 000240
1510 006124 000240
1511 006126 000240
1512 006130 000240
1513 006132 013702 177752
1514 006136 022702 000077
1515 006142 001021
1516 006144 000137 006222
1517

```

\*\*\*\*\*  
 TST6: SCOPE  
 MOV #40,\$TIMES ;;DO 40 ITERATIONS  
 KA=\$TN-1  
 MOV #TST7,SKAD ;SET THE SKAD REGISTER  
 ;IN CASE THE TEST ABORTS.  
 MOVB \$TSTNM,\$TMPO  
 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.  
 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.  
 CLR KAFLG  
 KA1: CLR @#CONTRL ;BOTH GROUPS ENABLED.  
 MOV #KA1,\$LPERR  
 MOV #KA2,R0  
 MOV #20,R1  
 KA2: TST (R0)+ ;SET UP HITS IN BOTH  
 SOB R1,KA2 ;GROUPS  
 NOP  
 NOP  
 NOP  
 MOV @#HITMIS,R2 ;SHOULD HAVE ALL HITS.  
 CMP #77,R2  
 BNE KAERR1  
 KA3: MOV #KA3,\$LPERR  
 MOV #S1M0,@#CONTRL ;DISABLE GROUP ZERO.  
 MOV #KA4,R0  
 MOV #20,R1  
 KA4: TST (R0)+ ;SET UP HITS IN GROUP 1  
 SOB R1,KA4  
 NOP  
 NOP  
 NOP  
 MOV @#HITMIS,R2 ;SHOULD HAVE ALL HITS.  
 CMP #77,R2  
 BNE KAERR2  
 KA5: MOV #KA5,\$LPERR  
 MOV #S0M1,@#CONTRL ;DISABLE GROUP ONE.  
 MOV #KA6,R0  
 MOV #20,R1  
 KA6: TST (R0)+ ;SET UP HITS IN GROUP ZERO.  
 SOB R1,KA6  
 NOP  
 NOP  
 NOP  
 MOV @#HITMIS,R2 ;SHOULD HAVE SIX HITS.  
 CMP #77,R2  
 BNE KAERR3  
 JMP KADONE

1518 006150 000000  
 1519  
 1520 006152  
 1521 006152 010237 001230  
 1522 006156 104067  
 1523 006160 052737 000001 006150  
 1524 006166 000714  
 1525 006170  
 1526 006170 010237 001230  
 1527 006174 104070  
 1528 006176 052737 000002 006150  
 1529 006204 000732  
 1530 006206  
 1531 006206 010237 001230  
 1532 006212 104071  
 1533 006214 052737 000004 006150  
 1534 006222 005037 177746  
 1535 006226 022737 000007 006150  
 1536 006234 001004  
 1537 006236 012737 177777 032332  
 1538 006244 000407  
 1539  
 1540 006246 032737 000006 006150  
 1541 006254 001403  
 1542 006256 012737 177777 032342  
 1543 006264  
 1544  
 1545  
 1546  
 1547  
 1548  
 1549  
 1550  
 1551  
 1552  
 1553  
 1554  
 1555  
 1556  
 1557  
 1558  
 1559  
 1560 006264 000004  
 1561 006266 012737 000040 001274  
 1562 000007  
 1563  
 1564 006274 012737 006614 032110  
 1565  
 1566 006302 113737 001102 001224  
 1567 006310 012737 031764 000114  
 1568  
 1569 006316 104432  
 1570 006320 104436  
 1571  
 1572 006322 012700 006612  
 1573 006326 042700 176003

KAFLG: .WORD 0 ;ERROR FLAG.  
 KAERR1: ;FAILED TO GET HITS  
 1\$: MOV R2,\$TMP2 ;WITH THE CONTROL  
 ERROR 67 ;REGISTER CLEAR!  
 BIS #BIT0,KAFLG  
 BR KA3  
 KAERR2: ;FAILED TO GET HITS  
 1\$: MOV R2,\$TMP2 ;WITH THE CONTROL REGISTER  
 ERROR 70 ;SET TO FORCE SELECT GROUP  
 BIS #BIT1,KAFLG ;ONE FORCE MISS GROUP ZERO.  
 BR KA5  
 KAERR3: ;FAILED TO GET HITS  
 1\$: MOV R2,\$TMP2 ;WITH THE CONTROL REGISTER  
 ERROR 71 ;SET TO FORCE SELECT GROUP  
 BIS #BIT2,KAFLG ;ZERO AND FORCE MISS GROUP ONE.  
 KADONE: CLR @#CONTRL  
 CMP #7,KAFLG ;IF THE TEST FAILED FOR ALL  
 BNE KAD2 ;THREE CONDITIONS OF THE  
 MOV #-1,HIMFLG ;CONTROL REGISTER SIGNAL  
 BR KAD3 ;A BAD HIT/MISS REGISTER.  
 KAD2: BIT #6,KAFLG ;IF THE TEST FAILED ONLY WHEN  
 BEQ KAD3 ;THE CONTROL REGISTER WAS SET  
 MOV #-1,CONFL2 ;SIGNAL A BAD CONTROL REGISTER.  
 KAD3: ;DONE!!

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

TST7: SCOPE  
 MOV #40,\$TIMES ;.DC 40 ITERATIONS  
 KD \$TN-1  
 MOV #TST10,SKAD ;SET THE SKAD REGISTER  
 ;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.  
 K1D: MOV #KTMP2D,R0 ;DETERMINE THE TEST LOCATIONS.  
 BIC #176003,R0



```

1574 006332 010001          MOV      R0,R1
1575 006334 062701 140000    ADD      #TSTR1,R1
1576 006340 010137 001244    MOV      R1,$TMP10
1577 006344 005037 001246    CLR      $TMP11
1578 006350 010002          MOV      R0,R2
1579 006352 062702 142000    ADD      #TSTR2,R2
1580 006356 010237 001250    MOV      R2,$TMP12
1581 006362 005037 001252    CLR      $TMP13
1582
1583 006366 012737 000044 177746 K2D:  MOV      #S1M0,@#CONTRL ;MAKE (R1) A HIT IN
1584 006374 005711          TST      (R1)           ;GROUP GRM.
1585 006376 005711          TST      (R1)
1586 006400 032737 000010 177752  BIT      #10,@#HITMIS
1587 006406 001007          BNE      K3D
1588
1589                                ;REPORT ERROR, UNABLE
1590 006410 012737 000001 001230  MOV      #1,$TMP2      ;GET A HIT IN GROUP GRM.
1591 006416 012737 000044 001232  MOV      #S1M0,$TMP3
1592 006424 104075          1$:  ERROR  75
1593
1594 006426 012703 000030          K3D:  MOV      #S0M1,R3
1595 006432 042703 000017          BIC      #17,R3
1596 006436 010337 177746          MOV      R3,@#CONTRL ;FORCE SELECT GROUP GRS.
1597 006442 005712          TST      (R2)           ;MAKE (R2) A HIT IN GROUP
1598 006444 005712          TST      (R2)           ;GRS.
1599 006446 032737 000010 177752  BIT      #10,@#HITMIS
1600 006454 001006          BNE      K4D
1601
1602                                ;IF NOT, ERROR UNABLE TO
1603                                ;GET A HIT IN GROUP 0
1603 006456 010337 001232          MOV      R3,$TMP3
1604 006462 104076          1$:  ERROR  76
1605 006464 012737 177777 032342  MOV      #-1,CONFL2
1606
1607 006472 005037 177746          K4D:  CLR      @#CONTRL    ;NOW MAKE SURE (R1) IS
1608 006476 000240          NOP                                ;FOR SCOPING WITH AN OSCILLOSCOPE.
1609 006500 005711          TST      (R1)           ;STILL A HIT IN GROUP
1610 006502 032737 000010 177752  BIT      #10,@#HITMIS ;1, THAT IS MAKE SURE
1611 006510 001010          BNE      K5D           ;GROUP 1 WASN'T WRITTEN
1612                                ;WHILE FORCE SELECTING
1613                                ;GROUP GRS.
1614 006512 012737 000001 001230  MOV      #1,$TMP2
1615 006520 012737 000000 001232  MOV      #0,$TMP3
1616 006526 104077          1$:  ERROR  77
1617 006530 000424          BR       K6D
1618 006532 012703 000044          K5D:  MOV      #S1M0,R3    ;NOW SEE IF YOU CAN
1619 006536 042703 000063          BIC      #63,R3        ;GET A MISS AT (R2)
1620 006542 010337 177746          MOV      R3,@#CONTRL ;BY FORCING MISSES
1621 006546 005712          TST      (R2)           ;TO GRS.
1622 006550 032737 000010 177752  BIT      #10,@#HITMIS
1623 006556 001411          BEQ      K6D
1624                                ;SHOULD BE A MISS,
1625                                ;OTHERWISE ERROR.
1625 006560 012737 000000 001230  MOV      #0,$TMP2
1626 006566 010337 001232          MOV      R3,$TMP3
1627 006572 104117          1$:  ERROR  117
1628 006574 012737 177777 032342  MOV      #-1,CONFL2
1629

```

1630 006602 005037 177746  
 1631 006606 000402  
 1632  
 1633 006610 000000  
 1634 006612 000000  
 1635  
 1636 006614  
 1637  
 1638  
 1639  
 1640  
 1641  
 1642  
 1643  
 1644  
 1645  
 1646  
 1647  
 1648  
 1649  
 1650  
 1651  
 1652  
 1653 006614 000004  
 1654 006616 012737 000040 001274  
 1655 000010  
 1656  
 1657 006624 012737 007144 032110  
 1658  
 1659 006632 113737 001102 001224  
 1660 006640 012737 031764 000114  
 1661  
 1662 006646 104432  
 1663 006650 104436  
 1664  
 1665 006652 012700 007142  
 1666 006656 042700 176003  
 1667 006662 010001  
 1668 006664 062701 140000  
 1669 006670 010137 001244  
 1670 006674 005037 001246  
 1671 006700 010002  
 1672 006702 062702 142000  
 1673 006706 010237 001250  
 1674 006712 005037 001252  
 1675  
 1676 006716 012737 000030 177746  
 1677 006724 005711  
 1678 006726 005711  
 1679 006730 032737 000010 177752  
 1680 006736 001007  
 1681  
 1682  
 1683 006740 012737 000000 001230  
 1684 006746 012737 000030 001232  
 1685 006754 104075

```

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.
:*
:*****
TST'0: SCOPE
      MOV #40,$TIMES ;:DO 40 ITERATIONS
KE-$TN-1
      MOV #TST11,SKAD ;SET THE SKAD REGISTER
      ;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

;REPORT ERROR, UNABLE
;GET A HIT IN GROUP GRM.
1$: MOV #0,$TMP2
      MOV #SOM1,$TMP3
      ERROR 75
    
```

```

1686
1687 006756 012703 000044      K3E:  MOV      #S1M0,R3
1688 006762 042703 000017      BIC      #17,R3
1689 006766 010337 177746      MOV      R3,@#CONTRL      ;FORCE SELECT GROUP GRS.
1690 006772 005712              TST      (R2)              ;MAKE (R2) A HIT IN GROUP
1691 006774 005712              TST      (R2)              ;GRS.
1692 006776 032737 000010 177752  BIT      #10,@#HITMIS
1693 007004 001006              BNE      K4E
1694
1695
1696 007006 010337 001232              MOV      R3,$TMP3
1697 007012 104076              1$:     ERROR      76
1698 007014 012737 177777 032342  MOV      #-1,CONFL2
1699
1700 007022 005037 177746      K4E:   CLR      @#CONTRL      ;NOW MAKE SURE (R1) IS
1701 007026 000240              NOP
1702 007030 005711              TST      (R1)              ;FOR SCOPING WITH AN OSCILLOSCOPE
1703 007032 032737 000010 177752  BIT      #10,@#HITMIS      ;STILL A HIT IN GROUP
1704 007040 001010              BNE      K5E                ;0, THAT IS MAKE SURE
1705
1706
1707 007042 012737 000000 001230  MOV      #0,$TMP2
1708 007050 012737 000001 001232  MOV      #1,$TMP3
1709 007056 104077              1$:     ERROR      77
1710 007060 000424              BR       K6E
1711 007062 012703 000030      K5E:   MOV      #S0M1,R3      ;NOW SEE IF YOU CAN
1712 007066 042703 000063      BIC      #63,R3            ;GET A MISS AT (R2)
1713 007072 010337 177746      MOV      R3,@#CONTRL      ;BY FORCING MISSES
1714 007076 005712              TST      (R2)              ;TO GRS.
1715 007100 032737 000010 177752  BIT      #10,@#HITMIS
1716 007106 001411              BEQ     K6E                ;SHOULD BE A MISS,
1717
1718 007110 012737 000001 001230  MOV      #1,$TMP2
1719 007116 010337 001232              MOV      R3,$TMP3
1720 007122 104117              1$:     ERROR      117
1721 007124 012737 177777 032342  MOV      #-1,CONFL2
1722
1723 007132 005037 177746      K6E:   CLR      @#CONTRL
1724 007136 000402              BR       K7E
1725
1726 007140 000000      KTMP1E: .WORD      0
1727 007142 000000      KTMP2E: .WORD      0
1728
1729 007144      K7L:   ;DONE.
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741

```

```

*****
*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
*BOTHS 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.

```

```

1742
1743
1744 007144 000004
1745 007146 012737 000020 001274
1746 000011
1747
1748 007154 012737 007754 032110
1749
1750 007162 113737 001102 001224
1751 007170 012737 031764 000114
1752
1753 007176 104432
1754 007200 104436
1755 007202 005037 007636
1756 007206 012737 000002 007640
1757 007214 012737 007230 001110
1758 007222 012737 007644 007642
1759
1760
1761
1762
1763 007230 012701 140000
1764 007234 012702 142000
1765 007240 012700 001000
1766 007244 012737 000030 177746
1767 007252 005721
1768 007254 012737 000044 177746
1769 00: ADD #2,R0
1781 007324 006302
1782 007326 103001
1783 007330 005710
1784 007332 062700 000006
1785 007336 077113
1786
1787 007340 012705 177752
1788 007344 000402
1789
1790
1791
1792
1793
1794
1795
1796 007346
1797 007344

;*****
TST11: SCOPE
MOV #20,$TIMES ;;DO 20 ITERATIONS
KC-$TN-1
MOV #ST12,SKAD ;SET THE SKAD REGISTER
;IN CASE THE TEST ABORTS.
MOVB $STNM,$TMP0
MOV #SPUR,@#CACHVEC
SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
SKPRHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
CLR KCCON ;TEST THE BOTH GROUPS
MOV #2,KCFLG1 ;ENABLED CONDITION FIRST.
KCO: MOV #KC1,$LPERR
MOV #KCTBL,KCPTR ;KCPTR IS A POINTER TO
;THE TABLE OF 12-BIT PATTERNS
;WHICH WILL BE FLOATED
;THROUGH THE REGISTER.
KC1: MOV #TESTR1,R1 ;MAKE THIS CODE MISSES
MOV #TESTR2,R2 ;TO BOTH GROUPS.
MOV #1000,R0
1$: MOV #SOM1,@#CONTRL
TST (R1)+
MOV #S1M0,@#CONTRL
ASL R2
BCC 1$
TST (R0) ;MAKE (R0) A HIT.
1$: ADD #6,R0
SOB R1,KC2
MOV #HITMIS,R5 ;NOW THAT THE HITS
BR KC3 ;AND MISSES HAVE BEEN
;APPROPRIATELY ESTABLISHED
;EXECUTE THE CODE AND
;CAUSE THE PATTERN TO FLOAT
;THROUGH THE HIT/MISS
;REGISTER.
LOC= ;GET THE PC TO AN EVEN WORD BOUNDARY...
LOC=-4&LOC
    
```

1798	007350			LOC-LOC+4		
1799	007350			.=LOC		
1800						
1801	007350	000000		HALT		
1802	007352	000240		KC3: NOP		:THE HALT'S HERE ARE NOT
1803	007354	000402		BR KC4		:EXECUTED, THEY ARE FILLERS.
1804	007356	000000		HALT		:THE ADDRESS OF THE HIT AND
1805	007360	000000		HALT		:MISS REGISTER IS IN R5.
1806	007362	011500		KC4: MOV (R5),R0		:NOTE THAT THE HIT/MISS
1807	007364	000402		BR KC5		:REGISTER IS READ EVERY
1808	007366	000000		HALT		:TWO CYCLES AND SAVED IN
1809	007370	000000		HALT		:A PROCESSOR GENERAL
1810	007372	011501		KC5: MOV (R5),R1		:PURPOSE REGISTER.
1811	007374	000402		BR KC6		
1812	007376	000000		HALT		
1813	007400	000000		HALT		
1814	007402	011502		KC6: MOV (R5),R2		
1815	007404	000402		BR KC7		
1816	007406	000000		HALT		
1817	007410	000000		HALT		
1818	007412	011503		KC7: MOV (R5),R3		
1819	007414	000402		BR KC8		
1820	007416	000000		HALT		
1821	007420	000000		HALT		
1822	007422	011504		KC8: MOV (R5),R4		
1823	007424	000402		BR KC9		
1824	007426	000000		HALT		
1825	007430	000000		HALT		
1826	007432	011505		KC9: MOV (R5),R5		:CAN SAVE PATTERN IN R5
1827						:SINCE THE ADDRESS IS
1828						:NO LONGER NEEDED.
1829	007434	042700	177774	KC10: BIC #177774,R0		:GET THE PATTERNS READ
1830	007440	010037	007670	MOV R0,KCRO		:FROM THE HIT/MISS REGISTER
1831	007444	042701	017760	BIC #17760,R1		:INTO LOCATIONS KCRO
1832	007450	010137	007672	MOV R1,KCR1		:THROUGH KCR5 SO THE
1833	007454	010237	007674	MOV R2,KCR2		:GENERAL PURPOSE REGISTERS
1834	007460	010337	007676	MOV R3,KCR3		:CAN BE USED FOR OTHER
1835	007464	010437	007700	MOV R4,KCR4		:THINGS
1836	007470	010537	007702	MOV R5,KCR5		
1837						
1838	007474	017701	000142	KC11: MOV @KCPTR,R1		
1839	007500	005000		CLR R0		
1840	007502	012702	000006	MOV #6,R2		:PUT THE EXPECTED VALUES
1841	007506	012703	007704	MOV #KCE0,R3		:IN KCE0 THROUGH KCE5.
1842	007512	073027	000002	KC12: ASHC #2,R0		
1843	007516	042700	177700	BIC #177700,R0		
1844	007522	010023		MOV R0,(R3)+		
1845	007524	077206		SOB R2,KC12		
1846						
1847	007526	012700	007670	MOV #KCR0,R0		
1848	007532	012701	007704	MOV #KCE0,R1		:MAKE SURE THE PATTERNS
1849	007536	012702	000006	MOV #6,R2		:WHICH WERE READ FROM
1850	007542	022021		KC13: CMP (R0)+,(R1)+		:THE HIT AND MISS REGISTER
1851	007544	001402		BEQ KC14		:MATCH THE EXPECTED
1852	007546	000137	007720	JMP KCERR		:PATTERNS.
1853	007552	077205		KC14: SOB R2,KC13		

```

1854
1855 007554 062737 000002 007642 KC15: ADD #2,KCPTR ;MOVE POINTER TO NEXT
1856 007562 023727 007642 007666 CMP KCPTR,#KCTBLB ;PATTERN AND IF ALL THE
1857 007570 001402 BEQ 1$ ;PATTERNS HAVEN'T BEEN
1858 007572 000137 007230 JMP KC1 ;TESTED GO TO KC1 TO TEST
1859 ;THIS NEXT PATTERN.
1860 007576 005337 007640 1$: DEC KCFLG1 ;IF ALL THE PATERNS HAVE BEEN
1861 007602 100002 BPL KC16 ;TESTED WITH THAT GROUP CONFIGURATION
1862 007604 000137 007750 JMP KCDONE ;SO GO TO THE NEXT CONFIGURATION.
1863 ;OR DONE!!
1864 007610 001405 KC16: BEQ KC17
1865 007612 012737 000044 007636 MOV #S1M0,KCCON ;BOTH GROUPS ENABLED CONFIGURATION
1866 007620 000137 007214 JMP KCO ;HAS BEEN TESTED SO NOW TEST GROUP
1867 ;ZERO DISABLED CONFIGURATION.
1868 007624 012737 000030 007636 KC17: MOV #S0M1,KCCON ;BOTH GROUPS ENABLED AND GROUP ZERO
1869 ;DISABLED CONFIGURATIONS HAVE BOTH
1870 ;BEEN TESTED SO FINALLY TEST THE
1871 007632 000137 007214 JMP KCO ;GROUP ONE D:ISABLED CONFIGURATION.
1872
1873
1874 007636 000000 KCCON: .WORD 0 ;PATTERN BEING USED IN THE CONTROL REGISTER
1875
1876 007640 000000 KCFLG1: .WORD 0 ;FLAG USED TO DETERMINE THE CONFIGURATION
1877 ;BEING TESTED.
1878 007642 000000 KCPTR: .WORD 0 ;POINTER USED TO POINT TO THE PATTERN
1879 ;BEING TESTED IN KCTBL.
1880
1881 007644 000000 KCTBL: .WORD 0 ;PATTERNS WHICH ARE
1882 007646 002000 .WORD 002000 ;FLOATED THROUGH THE HIT/MISS
1883 007650 177760 .WORD 177760 ;REGISTER. ONLY THE UPPER
1884 007652 175760 .WORD 175760 ;12 BITS HAVE ANY SIGNIFICANCE..
1885 007654 125240 .WORD 125240
1886 007656 146300 .WORD 146300
1887 007660 161600 .WORD 161600
1888 007662 100020 .WORD 100020
1889 007664 077740 .WORD 077740
1890 007666 000000 KCTBLB: .WORD 0
1891
1892 007670 000000 KCR0: .WORD 0 ;STORAGE FOR THE PATTERNS READ
1893 007672 000000 KCR1: .WORD 0 ;OUT OF THE HIT/MISS REGISTER.
1894 007674 000000 KCR2: .WORD 0
1895 007676 000000 KCR3: .WORD 0
1896 007700 000000 KCR4: .WORD 0
1897 007702 000000 KCR5: .WORD 0
1898
1899 007704 000000 KCE0: .WORD 0 ;EXPECTED VALUES FOR THE PATTERNS
1900 007706 000000 KCE1: .WORD 0 ;READ FROM THE HIT/MISS REGISTER.
1901 007710 000000 KCE2: .WORD 0
1902 007712 000000 KCE3: .WORD 0
1903 007714 000000 KCE4: .WORD 0
1904 007716 000000 KCE5: .WORD 0
1905
1906 007720 KCERR:
1907 007720 013737 007636 001230 MOV KCCON,$TMP2 ;REPORT THE PATTERN READ FROM THE
1908 007726 104120 1$: ERROR 120 ;HIT/MISS REGISTER WAS NOT THE EXPECTED
1909 007730 012737 177777 032342 MOV #-1,CONF2 ;VALUE.
    
```

```

1910 007736 012737 177777 032346
1911 007744 000137 007554
1912
1913 007750 005037 177746
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932 007754 000004
1933 000012
1934 007756 005737 032342
1935 007762 001403
1936 007764 012737 177777 032326
1937 007772 005737 032346
1938 007776 001403
1939 010000 012737 177777 032332
1940 010006
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955 010006 000004
1956 010010 012737 000040 001274
1957 000013
1958
1959 010016 012737 010242 032110
1960
1961 010024 113737 001102 001224
1962 010032 012737 031764 000114
1963
1964 010040 104432
1965 010042 104436
    
```

```

MOV # -1, HIMFL2
JMP KC15
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.
*
*****
TST12: SCOPE
KY $TN-1
TST CONFL2
BEQ KY1
MOV # -1, CONFLG
KY1: TST HIMFL2
BEQ KY2
MOV # -1, HIMFLG
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.
*
*****
TST13: SCOPE
MOV #40, $TIMES ;:DO 40 ITERATIONS
KF $TN-1
MOV #TST14, SKAD ;SET THE SKAD REGISTER
;IN CASE THE TEST ABORTS.
MOVB $TSTNM, $TMP0
MOV #SPUR, @#CACHVEC ;EXPECT NO PARITY ERRORS.
SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
    
```

1966	010044	012700	010240		KF1:	MOV	#KFTMP2,R0	:ESTABLISH A LOCATION FOR THE
1967								:HITS TO BE MADE WHICH WON'T
1968								:INTERFER WITH THE HITS CAUSED
1969								:BY EXECUTION OF THIS CODE.
1970	010050	042700	176003			BIC	#176003,R0	
1971	010054	010001				MOV	R0,R1	
1972	010056	062701	140000			ADD	#TESTR1,R1	
1973	010062	010002				MOV	R0,R2	
1974	010064	062702	142000			ADD	#TESTR2,R2	
1975								
1976	010070	012737	000044	177746		MOV	#S1M0,@#CONTRL	:MAKE THOSE TWO TEST LOCATIONS
1977	010076	005710				TST	(R0)	:(R1) AND (R2) MISSES IN BOTH
1978								:GROUPS BY MAKING (R0) A HIT
1979								:IN BOTH GROUPS.
1980								
1981	010100	005710				TST	(R0)	
1982								
1983								
1984	010102	032737	000010	177752		BIT	#10,@#HITMIS	:SEE IF REFERENCE ADDRESS
1985	010110	001006				BNE	KF2	:IS A HIT.
1986								:IF NOT ERROR!
1987	010112	010037	001230			MOV	R0,\$TMP2	
1988	010116	012737	000001	001226		MOV	#1,\$TMP1	
1989	010124	104001				ERROR	1	
1990								
1991								
1992								
1993								
1994	010126	012737	000030	177746	KF2:	MOV	#S0M1,@#CONTRL	
1995	010134	005710				TST	(R0)	
1996								
1997	010136	005710				TST	(R0)	
1998								
1999								
2000	010140	032737	000010	177752		BIT	#10,@#HITMIS	:SEE IF REFERENCE ADDRESS
2001	010146	001006				BNE	KF3	:IS A HIT.
2002								:IF NOT ERROR.
2003	010150	010037	001230			MOV	R0,\$TMP2	
2004	010154	012737	000000	001226		MOV	#0,\$TMP1	
2005	010162	104001				ERROR	1	
2006								
2007								
2008								
2009								
2010	010164	005037	177746		KF3:	CLR	@#CONTRL	:NOW THAT THE ADDRESSES (R1)
2011								:AND (R2) ARE MISSES, REFERENCING
2012								:THEM BOTH EACH IN CONSECUTIVE
2013								:REFERNCES SHOULD CAUSE THEM BOTH
2014								:TO BE MADE HITS IF THE RANDOM
2015								:FLIP FLOP TOGGLES INBETWEEN THE
2016								:TWO CYCLES.
2017								:NOTE THAT THESE TWO ADDRESSES
2018								:(R1) AND (R2) ARE SUCH THAT
2019								:IF THE RANDOM FLIP FLOP DIDN'T TOGGLE
2020								:THE HITS AT THE ADDRESSES
2021								:WOULD BE MUTUALLY EXCLUSIVE.



```

2022                                     ;THAT IS BOTH THESE ADDRESSES
2023                                     ;CAN'T BE HITS IN THE SAME GROUP.
2024
2025 010170 000240                       NOP
2026 010172 021112                       CMP      (R1),(R2)      ;FOR SCOPING WITH AN OSCILLOSCOPE.
2027                                     ;HERE BOTH THE OPERAND FETCHES
2028 010174 021112                       CMP      (R1),(R2)      ;SHOULD BE MISSES.
2029                                     ;HERE BOTH THE OPERAND FETCHES
2030 010176 013705 177752                 MOV      @WHITMIS,R5    ;SHOULD BE HITS.
2031 010202 005105                       COM      R5
2032 010204 032705 000014                 BIT      #14,R5        ;BOTH HITS ELSE ERROR.
2033 010210 001411                       BEQ      KF4
2034
2035 010212 010137 001230                 MOV      R1,$TMP2      ;REPORT THE ERROR.
2036 010216 005037 001232                 CLR      $TMP3
2037 010222 010237 001234                 MOV      R2,$TMP4
2038 010226 005037 001236                 CLR      $TMP5
2039
2040 010232 104121                       1$:      ERROR 121
2041 010234 000402                       KF4:     BR      KF5
2042
2043 010236 000000                       KFTMP1: .WORD 0        ;USED TO DETERMINE THE TEST
2044 010240 000000                       KFTMP2: .WORD 0        ;ADDRESSES.
2045
2046 010242                               KF5:     ;DONE.
2047
2048                                     ;*****
2049                                     ;*TEST 14      CACHE MAINTENANCE REGISTER COUNT PATTERN TEST
2050                                     ;*
2051                                     ;*THIS TEST RUNS A COUNT PATTERN THROUGH THE MAINTENANCE REGISTER'S
2052                                     ;*BITS 15 TO 4. THIS IS DONE TO INSURE THAT THESE BITS ARE SETABLE
2053                                     ;*AND THAT THE DATA PATH TO THE REGISTERS IS VIABLE. MISSES ARE FORCED
2054                                     ;*TO BOTH GROUPS SO THAT NO CACHE DATA OR ADDRESS MEMORY
2055                                     ;*ERRORS SHOULD OCCUR. ALSO ANY CYCLES DONE TO MAIN MEMORY
2056                                     ;*ARE INSURED, BY PROPER SELECTION OF INSTRUCTIONS, TO RETURN
2057                                     ;*DATA WITH THE PARITY BITS ON SO AS TO NOT CAUSE MAIN MEMORY PARITY
2058                                     ;*ERRORS BY SETTING THE MAIN MEMORY MAINTENANCE FUNCTION WHICH WOULD
2059                                     ;*EFFECTIVELY FORCE THE PARITY BITS READ FROM MAIN MEMORY TO A
2060                                     ;*ONE. SINCE THESE PARITY ARE ALREADY ONES, NO ERRORS SHOULD OCCUR.
2061                                     ;*
2062                                     ;*****
2063 010242 000004                       TST14:  SCOPE
2064 010244 012737 000020 001274           MOV      #20,$TIMES    ;;DO 20 ITERATIONS
2065 000014                               MA-$TN-1
2066
2067 010252 012737 010524 032110           MOV      #TST15,SKAD   ;SET THE SKAD REGISTER
2068                                     ;IN CASE THE TEST ABORTS.
2069 010260 113737 001102 001224           MOV8     $TSTNM,$TMP0
2070
2071 010266 104432                       SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2072 010270 104434                       SKPBMN      ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2073 010272 012737 010426 000114           MOV      #MAERR,@CACHVEC ;IN CASE AN ERROR OCCURS WHILE
2074                                     ;RUNNING A COUNT PATTERN
2075                                     ;THROUGH THE MAINTENANCE
2076                                     ;REGISTER SET UP THE PARITY ERROR
2077                                     ;TRAP VECTOR; NOTE THAT NO ERRORS
    
```

```

2078
2079
2080
2081 010300 012737 000014 177746      MOV      #MOM1,@#CONTRL
2082
2083 010306 012701 177750      MOV      #MAINT,R1
2084 010312 005004      CLR      R4
2085 010314 012737 010326 001110      MOV      #MA1,$LPERR
2086 010322 012700 170000      MOV      #170000,R0
2087
2088 010326 000240      MA1:    NOP
2089 010330 010411      MOV      R4,(R1)
2090 010332 011102      MOV      (R1),R2
2091 010334 005011      CLR      (R1)
2092
2093
2094
2095
2096 010336 030011      BIT      R0,(R1)
2097
2098
2099
2100
2101 010340 001402      BEQ      .+6
2102 010342 000000      HALT
2103
2104
2105
2106
2107
2108
2109
2110 010344 000240      MA2:    NOP
2111
2112 010346 011105      MOV      (R1),R5
2113 010350 001410      BEQ      MA3
2114
2115
2116 010352 010437 001230      MOV      R4,$TMP2
2117 010356 010537 001232      MOV      R5,$TMP3
2118 010362 104122      1$:    ERROR 122
2119 010364 012737 177777 032330      MOV      #-1,MANFLG
2120
2121 010372 020402      MA3:    CMP      R4,R2
2122 010374 001410      BEQ      MA4
2123
2124
2125 010376 010437 001230      MOV      R4,$TMP2
2126 010402 010237 001232      MOV      R2,$TMP3
2127 010406 104123      1$:    ERROR 123
2128 010410 012737 177777 032344      MOV      #-1,MANFL2
2129
2130 010416 062704 000020      MA4:    ADD      #20,R4
2131 010422 001341      BNE     MA1
2132 010424 000432      BR      MADONE
2133

```

```

:SHOULD OCCUR IF THIS REGISTER
:AND THE PARITY LOGIC IS FUNCTIONING
:PROPERLY.
:FORCE MISSES TO BOTH GROUPS.

:NOTE, THE CODE IN THIS ARE
:MA1 THROUGH MA2, ASSEMBLES TO
:MACHINE CODE WHICH WILL
:HAVE THE PARITY BITS ON, 1'S!
:THE PATTERN IS LOADED INTO THE
:MAINTENANCE REGISTER, READ BACK
:AND THE MAINTENANCE REGISTER
:IS CLEARED.
:SEE IF ANY OF THE HIGH ORDER
:FOUR BITS, 15 TO 12,
:THE BITS WHICH CONTROL THE
:MAIN MEMORY DATA PARITY MAINTENANCE
:FUNCTION ARE STUCK ON.
:IF SO, THEN ALL THAT CAN
:BE DONE IS TO HALT!!!!!!
:FOR IF CONTROL IS PASSED TO
:ANY OTHER PART OF THIS PROGRAM
:THERE WOULD BE NO CONTROL
:OVER WHAT KIND OF DATA WOULD
:BE READ FROM MAIN MEMORY AND
:MAIN MEMORY DATA PARITY ERRORS
:WOULD BE LIKELY TO OCCUR.

:SEE IF ANY OF THE LOW ORDER
:BITS, 11 THROUGH 0, ARE STUCK
:AT ONE.
:IF SO REPORT THE ERROR.

:SEE IF THE PATTERN WRITTEN MATCHES
:THE PATTERN READ.

:IF NOT REPORT THE ERROR.

:INCREMENT THE COUNT PATTERN.

```

2134	010426				MAERR:				:TRAP TO HERE IN THE EVENT
2135									:THAT A PARITY ERROR OCCURS
2136									:WHILE RUNNING THIS COUNT
2137									:PATTERN TEST.
2138	010426	032737	000400	177744		BIT	#400,@MEMERR		:SEE IF THE ERROR WAS A MAINTENANCE
2139	010434	001005				BNE	MAERR1		:ERROR, CAUSED BY A MAINTENANCE
2140									:FUNCTION. IF NOT GO TO THE
2141	010436	012737	031764	000114		MOV	#SPUR,@CACHVEC		:SPUR ROUTINE WHICH HANDLES SUCH UNEXPECTED

```
2142 010444 000137 031764          JMP      SPUR          :ERRORS.  
2143  
2144 010450 013737 177744 001234 MAERR1: MOV      @MEMERR,$TMP4 :IF THE ERROR WAS CAUSED BY A  
2145 010456 013737 177740 001226      MOV      @LOADRS,$TMP1 :MAINT FUNCTION THEN REPORT THE  
2146 010464 013737 177742 001230      MOV      @HIADRS,$TMP2 :FAILURE OF THAT REGISTER.  
2147 010472 012637 001232      MOV      (SP)+,$TMP3  
2148 010476 005726          TST      (SP)+
```

```

2149 010500 104124 1$: ERROR 124
2150 010502 012737 177777 032344 MOV #-1,MANFL2
2151
2152 010510 000742 BR MA4 ;RETURN TO THE TEST.
2153
2154 010512 005037 177746 MADONE: CLR @#CONTRL .DONE
2155 010516 012737 031764 000114 MOV #SPUR,@#CACHVEC
2156
2157
2158
2159
2160
2161 ::*****
2162 :*TEST 15 CACHE MAINTENANCE AND ERROR REGISTERS TEST 1
2163 :*
2164 :*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY
2165 :*ERROR ON THE MAIN MEMORY ADDRESS AND CONTROL LINES, AND ALSO A TEST
2166 :*OF THE ERROR REGISTER'S ABILITY TO APPROPRIATELY SET TO 104402. THE
2167 :*REFERENCE CAUSING THIS ERROR WILL BE MADE FROM THE CPU DIRECTLY TO
2168 :*THE CACHE.
2169 :*
2170 :*****
2171 010524 000004 TST15: SCOPE
2172 010526 012737 000040 001274 MOV #40,$TIMES ;:DO 40 ITERATIONS
2173 000015 MAB-$TN-1
2174 010534 012737 011022 032110 MOV #TST16,SKAD ;SET THE SKAD REGISTER
2175 ;IN CASE THE TEST ABORTS.
2176 010542 113737 001102 001224 MOVB $TSTNM,$TMP0
2177
2178 010550 104430 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2179 010552 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2180 010554 104434 SKPBMM ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2181 010556 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2182 010560 012737 010630 000114 MOV #MABRR0,@#CACHVEC ;SET UP FOR THE ERROR.
2183
2184 010566 012704 000002 MOV #2,R4 ;THIS IS THE PATTERN THAT WILL
2185 010572 012702 177750 MOV #MAINT,R2 ;BE PUT IN THE MAINTENANCE REG.
2186 010576 012737 000014 177746 MOV #MOM1,@#CONTRL ;FORCE MISSES TO BOTH GROUPS.
2187
2188 010604 000240 NOP ;FOR SCOPING.
2189 010606 010412 MOV R4,(R2) ;SET THE MAINTENANCE REGISTER.
2190 010610 005012 CLR (R2) ;THE REFERENCE WHICH FETCHES
2191 ;THIS INSTRUCTION SHOULD
2192 ;CAUSE THE ABORT!
2193
2194 010612 MAB2: ;NO ABORT OCCURRED REPORT THE ERROR
2195 010612 010437 001230 MOV R4,$TMP2
2196 010616 104127 1$: ERROR 127
2197 010620 012737 177777 032344 MOV #-1,MANFL2
2198 010626 000474 BR MABDON
2199
2200 010630 022737 104402 177744 MABRR0: CMP #104402,@#MEMERR ;WHEN THE TRAP IS MADE TO THIS LOCATION
2201 010636 001036 BNE MABRR4 ;MAKE SURE THE ERROR REGISTER IS
2202 ;SET CORRECTLY. IF NOT GO TO MABRR4.
2203 010640 022626 MABRR1: CMP (SP)+,(SP)+ ;OTHERWISE RESET THE STACK.
2204 010642 012737 177777 177744 MABRR15: MOV #-1,@#MEMERR ;ATTEMPT TO CLEAR THE ERROR REGISTER.
    
```

```

2205 010650 005737 177744      TST      @MEMERR
2206 010654 001416      BEQ      MABRR3
2207
2208 010656                    MABRR2:                    ;REPORT ERROR REGISTER WON'T CLEAR.
2209 010656 013737 177740 001230      MOV      @LOADRS,$TMP2
2210 010664 013737 177742 001232      MOV      @HIADRS,$TMP3
2211 010672 013737 177744 001234      MOV      @MEMERR,$TMP4
2212 010700 104130      1$:      ERROR      130
2213 010702 012737 177777 032324      MOV      #-1,MMRFLG
2214 010710 000443      BR       MABDON
2215
2216 010712 022737 177740 177740      MABRR3:  CMP      #177740,@LOADRS      ;MAKE SURE THE ADDRESS
2217 010720 001356      BNE      MABRR2                    ;REGISTER RESET.
2218 010722 022737 000003 177742      CMP      #3,@HIADRS
2219 010730 001352      BNE      MABRR2
2220 010732 000432      BR       MABDON
2221
2222 010734                    MABRR4:                    ;REPORT ERROR REGISTER NOT SET CORRECTLY!.
2223 010734 012637 001230      MOV      (SP)+,$TMP2
2224 010740 005726      TST      (SP)+
2225 010742 013737 177740 001232      MOV      @LOADRS,$TMP3
2226 010750 013737 177742 001234      MOV      @HIADRS,$TMP4
2227 010756 012737 000002 001236      MOV      #2,$TMP5
2228 010764 012737 104402 001240      MOV      #104402,$TMP6
2229 010772 013737 177744 001242      MOV      @MEMERR,$TMP7
2230 011000 104131      1$:      ERROR      131
2231 011002 012737 177777 032344      MOV      #-1,MANFL2
2232 011010 012737 177777 032340      MOV      #-1,MMRFL2
2233 011016 000711      BR       MABR15                    ;GO SEE IF THE ERROR REGISTER
2234                                     ;CAN BE CLEARED.
2235 011020 104416      MABDON:  RSET                    ;DONE..
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246 011022 000004      TST16:  SCOPE
2247 011024 012737 000040 001274      MOV      #40,$TIMES      ;;DO 40 ITERATIONS
2248                                     MB-$TN-1
2249                                     ;SET THE SKAD REGISTER
2250 011032 012737 011340 032110      MOV      #TST17,SKAD      ;IN CASE THE TEST ABORTS.
2251
2252 011040 113737 001102 001224      MOV      $TSTNM,$TMP0
2253
2254 011046 104430      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2255 011050 104432      SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2256 011052 104434      SKPBMM      ;IF THE MAINTENANCE REGISTER IS BAD SKIP THIS TEST.
2257 011054 104436      SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2258 011056 012737 011136 000114      MOV      #MBERR0,@CACHVEC      ;SET UP FOR THE ERROR.
2259 011064 012704 010000      MOV      #10000,R4            ;PATERN TO BE PUT INTO THE
2260 011070 012702 177750      MOV      #MAINT,R2           ;MAINTENANCE REGISTER.

```

```

:*****
:*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.
:*
:*****

```

```

2261 011074 012737 000014 177746 MOV #MOM1,@#CONTRL ;FORCE MISSES TO BOTH GROUPS.
2262 011102 000402 BR MB1
2263
2264 011104 LOC=. ;GET THE PC TO AN EVEN WORD BOUNDARY ..
2265 011104 LOC--4&LOC
2266 011110 LOC=LOC+4
2267 011110 .=LOC
2268
2269 011110 000240 MB1: NOP
2270 011112 010412 MOV R4,(R2) ;SET THE MAINTENANCE REGISTER.
2271 011114 005701 MB2: TST R1 ;THIS IS A DUMMY INSTRUCTION
2272 ;WITH THE APPROPRIATE PARITY
2273 ;WHOSE FETCH WILL CAUSE THE ERROR.
2274 011116 005012 CLR (R2)
2275
2276 011120 MB3:
2277 011120 010437 001230 MOV R4,$TMP2 ;REPORT ERROR. MAINTENANCE
2278 ;FUNCTION FAILED TO
2279 011124 104127 1$: ERROR 127 ;CAUSE ERROR.
2280 011126 012737 177777 032344 MOV #-1,MANFL2
2281 011134 000500 BR MBDONE
2282
2283 011136 022737 104404 177744 MBERR0: CMP #104404,@#MEMERR ;DID THE ERROR REGISTER
2284 011144 001042 BNE 69$ ;SET PROPERLY?
2285
2286 011146 022626 64$: CMP (SP)+,(SP)+ ;RESET THE STACK
2287 011150 005037 177572 65$: CLR @#MMR0
2288 011154 005037 172516 CLR @#MMR3
2289 011160 012737 177777 177744 MOV #-1,@#MEMERR ;TRY TO CLEAR THE ERROR
2290 011166 005737 177744 TST @#MEMERR ;REGISTER.
2291 011172 001416 BEQ 68$
2292
2293 011174 66$:
2294 011174 013737 177740 001230 MOV @#LOADRS,$TMP2 ;ERROR REGISTER WON'T
2295 011202 013737 177742 001232 MOV @#HIADRS,$TMP3 ;CLEAR
2296 011210 013737 177744 001234 MOV @#MEMERR,$TMP4
2297
2298 011216 104130 67$: ERROR 130
2299 011220 012737 177777 032324 MOV #-1,MMRFLG ;SIGNAL BAD REGISTER
2300 011226 000443 BR MBDONE
2301
2302 011230 022737 177740 177740 68$: CMP #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2303 011236 001356 BNE 66$ ;UNLOCKED.
2304 011240 022737 000003 177742 CMP #3,@#HIADRS
2305 011246 001352 BNE 66$
2306 011250 000432 BR MBDONE
2307
2308 011252 69$:
2309 011252 012637 001230 MOV (SP)+,$TMP2 ;REPORT ERROR REGISTER
2310 011256 005726 TST (SP)+ ;NOT SET AS EXPECTED.
2311 011260 013737 177740 001232 MOV @#LOADRS,$TMP3 ;RESET THE STACK.
2312 011266 013737 177742 001234 MOV @#HIADRS,$TMP4
2313 011274 012737 010000 001236 MOV #10000,$TMP5
2314 011302 012737 104404 001240 MOV #104404,$TMP6
2315 011310 013737 177744 001242 MOV @#MEMERR,$TMP7
2316
    
```

```

2317 011316 104131 70$: ERROR 131
2318 011320 012737 177777 032344 MOV #-1,MANFL2 ;SIGNAL BAD REGISTER
2319 011326 012737 177777 032340 MOV #-1,MMRF_2
2320 011334 000705 BR 65$
2321 011336 104416 MBDONE: RSET
2322
2323
2324 :*****
2325 :*TEST 17 CACHE MAINTENANCE AND ERROR REGISTERS TEST 3
2326 :*
2327 :*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2328 :*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S HIGH BYTE,
2329 :*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2330 :*
2331 011340 000004 TST17: SCOPE
2332 011342 012737 000040 001274 MOV #40,$TIMES ;;DO 40 ITERATIONS
2333 000017 MC=$TN-1
2334
2335 011350 012737 011654 032110 MOV #TST20,SKAD ;SET THE SKAD REGISTER
2336 ;IN CASE THE TEST ABORTS.
2337 011356 113737 001102 001224 MOVB $TSTNM,$TMP0
2338
2339 011364 104430 SKPBEP ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2340 011366 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2341 011370 104434 SKPBMM ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
2342 011372 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2343 011374 012737 011452 000114 MOV #MCERRO,@#CACHVEC ;SET UP FOR THE ERROR.
2344 011402 012704 020000 MOV #20000,R4 ;PATTERN TO BE USED IN THE
2345 011406 012702 177750 MOV #MAINT,R2 ;MAINTENANCE REGISTER.
2346 011412 012737 000014 177746 MOV #MOM1,@#CONTRL ;FORCE MISSES TO BOTH GROUPS.
2347 011420 000401 BR MC1
2348
2349 011422 LOC= ;GET THE PC TO AN EVEN WORD BOUNDARY..
2350 011420 LOC=-4&LOC
2351 011424 LOC=LOC+4
2352 011424 .=LOC
2353
2354 011424 000240 MC1: NOP
2355 011426 010412 MOV R4,(R2) ;SET THE MAINTENANCE REGISTER.
2356 011430 005701 MC2: TST R1 ;THE FETCH OF THIS INSTRUCTION
2357 ;SHOULD CAUSE THE ABORT.
2358 011432 005012 CLR (R2)
2359
2360 011434 MC3:
2361 011434 010437 001230 MOV R4,$TMP2 ;REPORT ERROR. MAINTENANCE
2362 ;FUNCTION FAILED TO
2363 ;CAUSE ERROR.
2363 011440 104127 1$: ERROR 127
2364 011442 012737 177777 032344 MOV #-1,MANFL2
2365 011450 000500 BR MCDONE
2366
2367 011452 022737 104404 177744 MCERRO: CMP #104404,@#MEMERR ;DID THE ERROR REGISTER
2368 011460 001042 BNE 69$ ;SET PROPERLY?
2369
2370 011462 022626 64$: CMP (SP)+,(SP)+ ;RESET THE STACK
2371 011464 005037 177572 65$: CLR @#MMR0
2372 011470 005037 172516 CLR @#MMR3

```



```

CEKBC-C PDP 11/70-74MP CACHE DIAGNOSTIC PART 1 MACY11 30A(1052) F 6 16-MAY-79 09:03 PAGE 47
CEKBCC.P11 02-MAY-79 15:02 T17 CACHE MAINTENANCE AND ERROR REGISTERS TEST 3 SEQ 0070

2373 011474 012737 177777 177744 MOV #-1,@MEMERR ;TRY TO CLEAR THE ERROR
2374 011502 005737 177744 TST @MEMERR ;REGISTER.
2375 011506 001416 BEQ 68$
2376
2377 011510 66$: MOV @LOADRS,$TMP2 ;ERROR REGISTER WON'T
2378 011510 013737 177740 001230 ;CLEAR
2379 011516 013737 177742 001232 MOV @HIADRS,$TMP3
2380 011524 013737 177744 001234 MOV @MEMERR,$TMP4
2381
2382 011532 104130 67$: ERROR 130
2383 011534 012737 177777 032324 MOV #-1,MMRFLG ;SIGNAL BAD REGISTER
2384 011542 000443 BR MCDONE
2385
2386 011544 022737 177740 177740 68$: CMP #177740,@LOADRS ;SEE IF ADDRESS REGISTER
2387 011552 001356 BNE 66$ ;UNLOCKED.
2388 011554 022737 000003 177742 CMP #3,@HIADRS
2389 011562 001352 BNE 66$
2390 011564 000432 BR MCDONE
2391
2392 011566 69$: MOV (SP)+,$TMP2 ;REPORT ERROR REGISTER
2393 011566 012637 001230 TST (SP)+ ;NOT SET AS EXPECTED.
2394 011572 005726 ;RESET THE STACK.
2395 011574 013737 177740 001232 MOV @LOADRS,$TMP3
2396 011602 013737 177742 001234 MOV @HIADRS,$TMP4
2397 011610 012737 020000 001236 MOV #20000,$TMP5
2398 011616 012737 104404 001240 MOV #104404,$TMP6
2399 011624 013737 177744 001242 MOV @MEMERR,$TMP7
2400
2401 011632 104131 70$: ERROR 131
2402 011634 012737 177777 032344 MOV #-1,MANFL2 ;SIGNAL BAD REGISTER
2403 011642 012737 177777 032340 MOV #-1,MMRFL2
2404 011650 000705 BR 65$
2405 011652 104416 MCDONE: RSET
2406
2407 ;*****
2408 ;*TEST 20 CACHE MAINTENANCE AND ERROR REGISTERS TEST 4
2409 ;*
2410 ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2411 ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE,
2412 ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2413 ;*
2414 ;*****
2415 011654 000004 TST20: SCOPE
2416 011656 012737 000040 001274 MOV #40,$T MES ;;DO 40 ITERATIONS
2417 000020 MD $TN-1
2418
2419 011664 012737 012174 032110 MOV #*ST21,SKAD ;SET THE SKAD REGISTER
2420 ;IN CASE THE TEST ABORTS.
2421 011672 113737 001102 001224 MOVB $STNM,$TMP0
2422
2423 011700 104430 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2424 011702 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2425 011704 104434 SKPBMM ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
2426 011706 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2427 011710 012737 011772 000114 MOV #MDERR0,@CACHVEC ;SET UP FOR THE ERROR.
2428 011716 012704 040000 MOV #40000,R4 ;PATTERN TO BE PUT IN THE

```

```

2429 011722 012702 177750      MOV      #MAINT,R2      ;MAINTENANCE REGISTER.
2430 011726 012737 000014 177746  MOV      #MOM1,@CONTRL ;FORCE MISSES TO BOTH GROUPS.
2431 011734 000402      BR       MD1
2432
2433          011736      LOC=      ;GET THE PC TO AN EVEN WORD BOUNDARY...
2434          011734      LOC=-4&LOC
2435          011740      LOC=LOC+4
2436          011740      .=LOC
2437
2438 011740 000240      NOP
2439 011742 000240      MD1:    NOP
2440 011744 010412      MD2:    MOV      R4,(R2)      ;SET THE MAINTENANCE REGISTER.
2441 011746 005701      TST     R1              ;THE FETCH OF THIS INSTRUCTION
2442                                     ;SHOULD CAUSE THE MAIN MEMORY
2443                                     ;DATA PARITY ABORT.
2444 011750 005012      CLR     (R2)
2445 011752 000240      NOP
2446
2447 011754          MD3:
2448 011754 010437 001230      MOV     R4,$TMP2      ;REPORT ERROR. MAINTENANCE
2449                                     ;FUNCTION FAILED TO
2450                                     ;CAUSE ERROR.
2451 011760 104127      1$:    ERROR    127
2452 011762 012737 177777 032344  MOV     #-1,MANFL2
2453 011770 000500      BR     MDDONE
2454 011772 022737 104410 177744  MDERR0: CMP     #104410,@MEMERR ;DID THE ERROR REGISTER
2455 012000 001042      BNE    69$            ;SET PROPERLY?
2456
2457 012002 022626      64$:   CMP     (SP)+,(SP)+ ;RESET THE STACK
2458 012004 005037 177572      65$:   CLR     @MMR0
2459 012010 005037 172516      CLR     @MMR3
2460 012014 012737 177777 177744  MOV     #-1,@MEMERR ;TRY TO CLEAR THE ERROR
2461 012022 005737 177744      TST     @MEMERR      ;REGISTER.
2462 012026 001416      BEQ    68$
2463
2464 012030          66$:
2465 012030 013737 177740 001230      MOV     @LOADRS,$TMP2 ;ERROR REGISTER WON'T
2466 012036 013737 177742 001232      MOV     @HIADRS,$TMP3 ;CLFAR
2467 012044 013737 177744 001234      MOV     @MEMERR,$TMP4
2468
2469 012052 104130      67$:   ERROR    130
2470 012054 012737 177777 032324  MOV     #-1,MMRFLG ;SIGNAL BAD REGISTER
2471 012062 000443      BR     MDDONE
2472
2473 012064 022737 177740 177740  68$:   CMP     #177740,@LOADRS ;SEE IF ADDRESS REGISTER
2474 012072 001356      BNE    66$            ;UNLOCKED.
2475 012074 022737 000003 177742  CMP     #3,@HIADRS
2476 012102 001352      BNE    66$
2477 012104 000432      BR     MDDONE
2478
2479 012106          69$:
2480 012106 012637 001230      MOV     (SP)+,$TMP2 ;REPORT ERROR REGISTER
2481 012112 005726      TST     (SP)+        ;NOT SET AS EXPECTED.
2482 012114 013737 177740 001232      MOV     @LOADRS,$TMP3 ;RESET THE STACK.
2483 012122 013737 177742 001234      MOV     @HIADRS,$TMP4
2484 012130 012737 040000 001236      MOV     #40000,$TMP5
    
```

```

2485 012136 012737 104410 001240      MOV      #104410,$TMP6
2486 012144 013737 177744 001242      MOV      @MEMERR,$TMP7
2487
2488 012152 104131      70$:    ERROR  131
2489 012154 012737 177777 032344      MOV      #-1,MANFL2      ;SIGNAL BAD REGISTER
2490 012162 012737 177777 032340      MOV      #-1,MMRFL2
2491 012170 000705      BR      65$
2492 012172 104416      MDDONE: RSET
2493
2494
2495      ;*****
2496      ;*TEST 21      CACHE MAINTENANCE AND ERROR REGISTERS TEST 5
2497      ;*
2498      ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2499      ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S HIGH BYTE,
2500      ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2501      ;*
2502      ;*****
2502 012174 000004      TST21:  SCOPE
2503 012176 012737 000040 001274      MOV      #40,$TIMES      ;;DO 40 ITERATIONS
2504      000021      ME=$TN-1
2505
2506 012204 012737 012514 032110      MOV      #TST22,SKAD      ;SET THE SKAD REGISTER
2507      ;IN CASE THE TEST ABORTS.
2508 012212 113737 001102 001224      MOV      $TSTNM,$TMP0
2509
2510 012220 104430      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2511 012222 104432      SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2512 012224 104434      SKPBMN      ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2513 012226 104436      SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2514 012230 012737 012312 000114      MOV      #MEERRO,@#CACHVEC      ;SET UP FOR THE ERROR.
2515 012236 012704 100000      MOV      #100000,R4      ;PATTERN TO BE PUT IN THE
2516 012242 012702 177750      MOV      #MAINT,R2      ;MAINTENANCE REGISTER.
2517 012246 012737 000014 177746      MOV      #MOM1,@#CONTRL      ;FORCE MISSES TO BOTH GROUPS.
2518 012254 000402      BR      ME1
2519
2520      012256      LOC=.      ;GET THE PC TO AN EVEN WORD BOUNDARY !!
2521      012254      LOC=-4&LOC
2522      012260      LOC=LOC+4
2523      012260      .=LOC
2524
2525 012260 000240      NOP
2526 012262 000240      ME1:    NOP
2527 012264 010412      MOV      R4,(R2)      ;SET THE MAINTENANCE REGISTER.
2528 012266 005701      ME2:    TST      R1      ;THE FETCH OF THIS INSTRUSTION
2529      ;SHOULD CAUSE THE ABORT.
2530 012270 005012      (LR      (R2)
2531 012272 000240      NOP
2532
2533 012274      ME3:
2534 012274 010437 001230      MOV      R4,$TMP2      ;REPORT ERROR. MAINTENANCE
2535      ;FUNCTION FAILED TO
2536      ;CAUSE ERROR.
2536 012300 104127      1$:    ERROR  127
2537 012302 012737 177777 032344      MOV      #-1,MANFL2
2538 012310 000500      BR      MEDONE
2539
2540 012312 022737 104410 177744      MEERRO: CMP      #104410,@MEMERR      ;DID THE ERROR REGISTER
    
```

```

2541 012320 001042      BNE      69$      ;SET PROPERLY?
2542
2543 012322 022626      64$:  CMP      (SP)+,(SP)+ ;RESET THE STACK
2544 012324 005037 177572      65$:  CLR      @MMR0
2545 012330 005037 172516      CLR      @MMR3
2546 012334 012737 177777 177744      MOV      #-1,@MEMERR ;TRY TO CLEAR THE ERROR
2547 012342 005737 177744      TST      @MEMERR      ;REGISTER.
2548 012346 001416      BEQ      68$
2549
2550 012350      66$:  MOV      @LOADRS,$TMP2 ;ERROR REGISTER WON'T
2551 012350 013737 177740 001230      MOV      @HIADRS,$TMP3 ;CLEAR
2552 012356 013737 177742 001232      MOV      @MEMERR,$TMP4
2553 012364 013737 177744 001234
2554
2555 012372 104130      67$:  ERROR    130
2556 012374 012737 177777 032324      MOV      #-1,MMRFLG ;SIGNAL BAD REGISTER
2557 012402 000443      BR       MEDONE
2558
2559 012404 022737 177740 177740      68$:  CMP      #177740,@LOADRS ;SEE IF ADDRESS REGISTER
2560 012412 001356      BNE      66$      ;UNLOCKED.
2561 012414 022737 000003 177742      CMP      #3,@HIADRS
2562 012422 001352      BNE      66$
2563 012424 000432      BR       MEDONE
2564
2565 012426      69$:  MOV      (SP)+,$TMP2 ;REPORT ERROR REGISTER
2566 012426 012637 001230      TST      (SP)+ ;NOT SET AS EXPECTED.
2567 012432 005726      MOV      @LOADRS,$TMP3 ;RESET THE STACK.
2568 012434 013737 177740 001232      MOV      @HIADRS,$TMP4
2569 012442 013737 177742 001234      MOV      #100000,$TMP5
2570 012450 012737 100000 001236      MOV      #104410,$TMP6
2571 012456 012737 104410 001240      MOV      @MEMERR,$TMP7
2572 012464 013737 177744 001242
2573
2574 012472 104131      70$:  ERROR    131
2575 012474 012737 177777 032344      MOV      #-1,MANFL2 ;SIGNAL BAD REGISTER
2576 012502 012737 177777 032340      MOV      #-1,MMRFL2
2577 012510 000705      BR       65$
2578 012512 104416      MEDONE: RSET
2579
2580      ;*****
2581      ;*TEST 22      CACHE MAINTENANCE AND ERROR REGISTERS TEST 6
2582      ;*
2583      ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2584      ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE,
2585      ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2586      ;*
2587      ;*****
2588 012514 000004      TST22: SCOPE
2589 012516 012737 000040 001274      MOV      #40,$TIMES ;:DO 40 ITERATIONS
2590      MF=$TN-1
2591
2592 012524 012737 013030 032110      MOV      #TST23,SKAD ;SET THE SKAD REGISTER
2593      ;IN CASE THE TEST ABORTS.
2594 012532 113737 001102 001224      MOV      $STNM,$TMP0
2595 012540 012737 012626 000114      MOV      @MFERR0,@CACHVEC ;SET UP FOR THE ERROR.
2596 012546 012704 010000      MOV      #10000,R4 ;PATTERN TO BE LOADED INTO THE

```

```

2597 012552 012702 177750      MOV      #MAINT,R2      ;MAINTENANCE REGISTER.
2598 012556 012737 000014 177746  MOV      #MOM1,@CONTRL ;FORCE MISSES TO BOTH GROUPS.
2599 012564 012705 012606      MOV      #MF2,R5       ;A REFERENCE TO THIS ADDRESS
2600                                     ;WILL CAUSE A PARITY TRAP BECAUSE
2601                                     ;THE OTHER WORD IN THE PAIR
2602                                     ;WILL HAVE THE APPROPRIATE
2603                                     ;PARITY TO CAUSE THE MAINTENANCE
2604                                     ;FUNCTION WHICH WILL BE SET
2605                                     ;TO FORCE THE ERROR.
2606 012570 000401      BR        MF1
2607
2608                                     LOC=.                  ;GET THE PC TO AN EVEN WORD BOUNDARY.
2609                                     LOC=-4&LOC
2610                                     LOC=LOC+4
2611                                     .=LOC
2612
2613 012574 000240      MF1:    NOP
2614 012576 010412      MOV      R4,(R2)       ;SET THE MAINTENANCE REGISTER.
2615 012600 021502      CMP      (R5),R2       ;THIS REFERENCE TO (R5) WILL CAUSE A
2616 012602 005012      CLR      (R2)         ;PARITY TRAP SINCE THE OTHER IN THAT
2617                                     ;PAIR WILL CAUSE A PARITY ERROR.
2618 012604 005701      TST      R1           ;THIS WORD WILL CAUSE THE ERROR
2619 012606 000240      MF2:    NOP           ;WHEN THIS WORD IS REFERENCED.
2620
2621 012610      MF3:
2622 012610 010437 001230      MOV      R4,$TMP2     ;REPORT ERROR. MAINTENANCE
2623                                     ;FUNCTION FAILED TO
2624                                     ;CAUSE ERROR.
2624 012614 104127      1$:    ERROR 127
2625 012616 012737 177777 032344  MOV      #-1,MANFL2
2626 012624 000500      BR        MFDONE
2627
2628 012626 022737 004404 177744  MFERR0: CMP      #4404,@MEMERR ;DID THE ERROR REGISTER
2629 012634 001042      BNE                                     ;SET PROPERLY?
2630
2631 012636 022626      64$:   CMP      (SP)+,(SP)+ ;RESET THE STACK
2632 012640 005037 177572      65$:   CLR      @MMR0
2633 012644 005037 172516      CLR      @MMR3
2634 012650 012737 177777 177744  MOV      #-1,@MEMERR ;TRY TO CLEAR THE ERROR
2635 012656 005737 177744      TST      @MEMERR     ;REGISTER.
2636 012662 001416      BEQ     68$
2637
2638 012664      66$:
2639 012664 013737 177740 001230      MOV      @LOADRS,$TMP2 ;ERROR REGISTER WON'T
2640 012672 013737 177742 001232      MOV      @HIADRS,$TMP3 ;CLEAR
2641 012700 013737 177744 001234      MOV      @MEMERR,$TMP4
2642
2643 012706 104130      67$:   ERROR 130
2644 012710 012737 177777 032324  MOV      #-1,MMRFLG   ;SIGNAL BAD REGISTER
2645 012716 000443      BR        MFDONE
2646
2647 012720 022737 177740 177740  68$:   CMP      #177740,@LOADRS ;SEE IF ADDRESS REGISTER
2648 012726 001356      BNE                                     ;UNLOCKED.
2649 012730 022737 000003 177742      CMP      #3,@HIADRS
2650 012736 001352      BNE     66$
2651 012740 000432      BR        MFDONE
2652
    
```

```

2653 012742                                69$: ;REPORT ERROR REGISTER
2654 012742 012637 001230                MOV (SP)+,$TMP2 ;NOT SET AS EXPECTED.
2655 012746 005726                        TST (SP)+ ;RESET THE STACK.
2656 012750 013737 177740 001232        MOV @#LOADRS,$TMP3
2657 012756 013737 177742 001234        MOV @#HIADRS,$TMP4
2658 012764 012737 010000 001236        MOV #10000,$TMP5
2659 012772 012737 004404 001240        MOV #4404,$TMP6
2660 013000 013737 177744 001242        MOV @#MEMERR,$TMP7
2661
2662 013006 104131                          70$: ERROR 131
2663 013010 012737 177777 032344        MOV #-1,MANFL2 ;SIGNAL BAD REGISTER
2664 013016 012737 177777 032340        MOV #-1,MMRFL2
2665 013024 000705                          BR 65$
2666 013026 104416                          MFDONE: RSET
2667
2668 ;*****
2669 ;*TEST 23 CACHE MAINTENANCE AND ERROR REGISTERS TEST 7
2670 ;*
2671 ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2672 ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE,
2673 ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2674 ;*
2675 ;*****
2676 013030 000004                          TST23: SCOPE
2677 013032 012737 000040 001274        MOV #40,$TIMES ;:DO 40 ITERATIONS
2678 000023                          MG $TN-1
2679
2680 013040 012737 013350 032110        MOV #TST24,SKAD ;SET THE SKAD REGISTER
2681 ;IN CASE THE TEST ABORTS.
2682 013046 113737 001102 001224        MOVB $TSTNM,$TMP0
2683
2684 013054 104430                          SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2685 013056 104432                          SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2686 013060 104434                          SKPBMMN ;IF THE MAINTENANCE REGISTER IS BAD SKIP THIS TEST.
2687 013062 104436                          SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2688 013064 012704 040000                MOV #40000,R4 ;THIS PATTERN WILL BE PUT IN THE
2689 013070 012702 177750                MOV #MAINT,R2 ;MAINTENANCE REGISTER.
2690 013074 012737 013146 000114        MOV #MGERR0,@#CACHVEC ;SET UP FOR THE ERROR.
2691 013102 012737 000014 177746        MOV #MOM1,@#CONTRL ;FORCE MISSES TO BOTH GROUPS.
2692 013110 000401                          BR MG1
2693
2694 013112                          LOC=. ;GET THE PC TO AN EVEN WORD BOUNDARY...
2695 013110                          LOC=-4&LOC
2696 013114                          LOC=LOC+4
2697 013114                          .=LOC
2698
2699 013114 000240                          MG1: NOP
2700 013116 010412                          MOV R4,(R2) ;SET THE MAINTENANCE REGISTER.
2701 013120 000240                          NOP ;THE REFERENCE TO THIS NOP
2702 013122 005701                          MG2: TST R1 ;SHOULD CAUSE A PARITY ERROR TO OCCUR AT
2703 ;MG2, RESULTING IN A TRAP!
2704 013124 005012                          CLR (R2)
2705 013126 000240                          NOP
2706
2707 013130                          MG3:
2708 013130 010437 001230                MOV R4,$TMP2 ;REPORT ERROR. MAINTENANCE
;FUNCTION FAILED TO
    
```

```

2709                                     ;CAUSE ERROR.
2710 013134 104127 1$: ERROR 127
2711 013136 012737 177777 032344 MOV #-1,MANFL2
2712 013144 000500 BR MGDONE
2713
2714 013146 022737 004410 177744 MGERR0: CMP #4410,@MEMERR ;DID THE ERROR REGISTER
2715 013154 001042 BNE 69$ ;SET PROPERLY?
2716
2717 013156 022626 64$: CMP (SP)+,(SP)+ ;RESET THE STACK
2718 013160 005037 177572 65$: CLR @MMR0
2719 013164 005037 172516 CLR @MMR3
2720 013170 012737 177777 177744 MOV #-1,@MEMERR ;TRY TO CLEAR THE ERROR
2721 013176 005737 177744 TST @MEMERR ;REGISTER.
2722 013202 001416 BEQ 68$
2723
2724 013204 66$: ;ERROR REGISTER WON'T
2725 013204 013737 177740 001230 MOV @LOADRS,$TMP2 ;CLEAR
2726 013212 013737 177742 001232 MOV @HIADRS,$TMP3
2727 013220 013737 177744 001234 MOV @MEMERR,$TMP4
2728
2729 013226 104130 67$: ERROR 130
2730 013230 012737 177777 032324 MOV #-1,MMRFLG ;SIGNAL BAD REGISTER
2731 013236 000443 BR MGDONE
2732
2733 013240 022737 177740 177740 68$: CMP #177740,@LOADRS ;SEE IF ADDRESS REGISTER
2734 013246 001356 BNE 66$ ;UNLOCKED.
2735 013250 022737 000003 177742 CMP #3,@HIADRS
2736 013256 001352 BNE 66$
2737 013260 000432 BR MGDONE
2738
2739 013262 69$: ;REPORT ERROR REGISTER
2740 013262 012637 001230 MOV (SP)+,$TMP2 ;NOT SET AS EXPECTED.
2741 013266 005726 TST (SP)+ ;RESET THE STACK.
2742 013270 013737 177740 001232 MOV @LOADRS,$TMP3
2743 013276 013737 177742 001234 MOV @HIADRS,$TMP4
2744 013304 012737 040000 001236 MOV #40000,$TMP5
2745 013312 012737 004410 001240 MOV #4410,$TMP6
2746 013320 013737 177744 001242 MOV @MEMERR,$TMP7
2747
2748 013326 104131 70$: ERROR 131
2749 013330 012737 177777 032344 MOV #-1,MANFL2 ;SIGNA BAD REGISTER
2750 013336 012737 177777 032340 MOV #-1,MMRFL2
2751 013344 000705 BR 65$
2752 013346 104416 MGDONE: RSET
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
    
```

```

:*****
:*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.
:*
    
```

```

2765
2766 013350 000004
2767 013352 012737 000040 001274
2768 000024
2769
2770 013360 012737 013714 032110
2771
2772 013366 113737 001102 001224
2773
2774 013374 104430
2775 013376 104432
2776 013400 104434
2777 013402 104436
2778 013404 012737 013512 000114
2779 013412 012704 000400
2780 013416 012702 177750
2781 013422 012737 000030 177746
2782
2783
2784 013430 012705 013472
2785 013434 005715
2786 013436 005715
2787
2788
2789 013440 032737 000010 177752
2790 013446 001007
2791
2792 013450 010537 001230
2793 013454 012737 000000 001226
2794 013462 104001
2795
2796 013464 104420
2797
2798 013466 000240
2799 013470 010412
2800 013472 005012
2801
2802
2803
2804
2805
2806 013474
2807 013474 010437 001230
2808
2809 013500 104127
2810 013502 012737 177777 032344
2811 013510 000500
2812
2813 013512 022737 004420 177744
2814 013520 001042
2815
2816 013522 022626
2817 013524 005037 177572
2818 013530 005037 172516
2819 013534 012737 177777 177744
2820 013542 005737 177744

```

\*\*\*\*\*  
 TST24: SCOPE  
 MOV #40,\$TIMES ;:DO 40 ITERATIONS  
 MH=\$TN-1  
 .OV #TST25,SKAD ;SET THE SKAD REGISTER  
 ;IN CASE THE TEST ABORTS.  
 MOVB \$TSTNM,\$TMPO  
 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.  
 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.  
 SKPBMM ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.  
 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.  
 MOV #MHERR0,@#CACHVEC ;SET UP FOR THE ERROR.  
 MOV #400,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 #MH1,R5 ;MAKE MH1 A HIT IN  
 TST (R5) ;GROUP GP.  
 TST (R5)  
 BIT #10,@#HITMIS ;SEE IF REFERENCE ADDRESS  
 BNF 1\$ ;IS A HIT.  
 ;IF NOT ERROR.  
 MOV R5,\$TMP2  
 MOV #0,\$TMP1  
 ERROR 1  
 SKIPT ;ERROR FATAL. GO TO NEXT TEST.  
 1\$: NOP ;PUT THE PATTERN IN THE  
 MOV R4,(R2) ;MAINTENANCE REGISTER.  
 MH1: CLR (R2) ;THE FETCH OF THIS NEXT  
 ;INSTRUCTION SHOULD CAUSE  
 ;A PARITY ERROR IN THE  
 ;CACHE ADDRESS MEMORY GROUP GP.  
 MH2: ;REPORT ERROR. MAINTENANCE  
 MOV R4,\$TMP2 ;FUNCTION FAILED TO  
 ;CAUSE ERROR.  
 1\$: ERROR 127  
 MOV #-1,MANFL2  
 BR MHDONE  
 MHERR0: CMP #4420,@#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.



```

2821 013546 001416 BEQ 68$
2822
2823 013550 66$:
2824 013550 013737 177740 001230 MOV @#LOADRS,$TMP2 ;ERROR REGISTER WON'T
2825 013556 013737 177742 001232 MOV @#HIADRS,$TMP3 ;CLEAR
2826 013564 013737 177744 001234 MOV @#MEMERR,$TMP4
2827
2828 013572 104130 67$: ERROR 130
2829 013574 012737 177777 032324 MOV #-1,$MMRFLG ;SIGNAL BAD REGISTER
2830 013602 000443 BR M#DONE
2831
2832 013604 022737 177740 177740 68$: CMP #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2833 013612 001356 BNE 66$ ;UNLOCKED.
2834 013614 022737 000003 177742 CMP #3,@#HIADRS
2835 013622 001352 BNE 66$
2836 013624 000432 BR M#DONE
2837
2838 013626 69$:
2839 013626 012637 001230 MOV (SP)+,$TMP2 ;REPORT ERROR REGISTER
2840 013632 005726 TST (SP)+ ;NOT SET AS EXPECTED.
2841 013634 013737 177740 001232 MOV @#LOADRS,$TMP3 ;RESET THE STACK.
2842 013642 013737 177742 001234 MOV @#HIADRS,$TMP4
2843 013650 012737 000400 001236 MOV #400,$TMP5
2844 013656 012737 004420 001240 MOV #4420,$TMP6
2845 013664 013737 177744 001242 MOV @#MEMERR,$TMP7
2846
2847 013672 104131 70$: ERROR 131
2848 013674 012737 177777 032344 MOV #-1,$MANFL2 ;SIGNAL BAD REGISTER
2849 013702 012737 177777 032340 MOV #-1,$MMRFL2
2850 013710 000705 BR 65$
2851 013712 104416 *#DONE: RSET
2852
2853
2854
2855 ;*****
2856 ;*TEST 25 CACHE MAINTFNANCE AND ERROR REGISTERS TEST 11
2857 ;*
2858 ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
2859 ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO, FOR THE
2860 ;*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
2861 ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
2862 ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
2863 ;*TO THE CACHE.
2864 ;*
2865 ;*****
2866 TST25: SCOPE
2867 MOV #40,$TIMES ;:DO 40 ITERATIONS
2868 MI-$TN-1
2869 MOV #TST26,$KAD ;SET THE SKAD REGISTER
2870 ;:IN CASE THE TEST ABORTS.
2871 MOV $TSTNM,$TMP0
2872
2873 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2874 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2875 SKPBMN ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2876 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
    
```

```

2877 013750 012737 014056 000114      MOV    #MIERRO,@#CACHVEC      ;SET UP FOR THE ERROR.
2878 013756 012704 001000              MOV    #1000,R4                ;PATTERN TO BE PUT IN MAINT. REG.
2879 013762 012702 177750              MOV    #MAINT,R2
2880 013766 012737 000030 177746      MOV    #SOM1,@#CONTRL        ;FORCE SELECT GROUP 0 AND
2881                                ;FORCE MISS THE OTHER
2882                                ;GROUP
2883 013774 012705 014036              MOV    #MI1,R5                ;MAKE MI1 A HIT IN
2884 014000 005715                      TST    (R5)                   ;GROUP GP.
2885 014002 005715                      TST    (R5)
2886
2887                                ;SEE IF REFERENCE ADDRESS
2888 014004 032737 000010 177752      BIT    #10,@#HITMIS          ;IS A HIT.
2889 014012 001007                      BNE    1$
2890                                ;IF NOT ERROR.
2891 014014 010537 001230              MOV    R5,$TMP2
2892 014020 012737 000000 001226      MOV    #0,$TMP1
2893 014026 104001                      ERROR  1
2894
2895 014030 104420                      SKIPT
2896                                ;ERROR FATAL. GO TO NEXT TEST.
2897                                1$:
2897 014032 000240                      NOP
2898 014034 010412                      MOV    R4,(R2)
2899 014036 005012                      MI1:  CLR    (R2)
2900                                ;PUT THE PATTERN IN THE
2901                                ;MAINTENANCE REGISTER.
2902                                ;THE FETCH OF THIS NEXT
2903                                ;INSTRUCTION SHOULD CAUSE
2904                                ;A PARITY ERROR IN THE
2905                                ;CACHE ADDRESS MEMORY GROUP GP.
2905 014040                                MI2:
2906 014040 010437 001230              MOV    R4,$TMP2
2907                                ;REPORT ERROR. MAINTENANCE
2908                                ;FUNCTION FAILED TO
2909                                ;CAUSE ERROR.
2908 014044 104127                      1$:  ERROR  127
2909 014046 012737 177777 032344      MOV    #-1,MANFL2
2910 014054 000500                      BR     MIDONE
2911
2912 014056 022737 004420 177744      MIERRO: CMP    #4420,@#MEMERR    ;DID THE ERROR REGISTER
2913 014064 001042                      BNE    69$                    ;SET PROPERLY?
2914
2915 014066 022626                      64$:  CMP    (SP)+,(SP)+        ;RESET THE STACK
2916 014070 005037 177572                      65$:  CLR    @#MMR0
2917 014074 005037 172516                      CLR    @#MMR3
2918 014100 012737 177777 177744      MOV    #-1,@#MEMERR          ;TRY TO CLEAR THE ERROR
2919 014106 005737 177744                      TST    @#MEMERR              ;REGISTER.
2920 014112 001416                      BEQ    68$
2921
2922 014114                                66$:
2923 014114 013737 177740 001230      MOV    @#LOADRS,$TMP2        ;ERROR REGISTER WON'T
2924 014122 013737 177742 001232      MOV    @#HIADRS,$TMP3        ;CLEAR
2925 014130 013737 177744 001234      MOV    @#MEMERR,$TMP4
2926
2927 014136 104130                                67$:  ERROR  130
2928 014140 012737 177777 032324      MOV    #-1,MMRFLG           ;SIGNAL BAD REGISTER
2929 014146 000443                      BR     MIDONE
2930
2931 014150 022737 177740 177740      68$:  CMP    #177740,@#LOADRS    ;SEE IF ADDRESS REGISTER
2932 014156 001356                      BNE    66$                  ;UNLOCKED.
    
```

CE  
CE

```

2933 014160 022737 000003 177742      CMP      #3,@#HIADRS
2934 014166 001352                      BNE      66$
2935 014170 000432                      BR       MIDONE
2936
2937 014172                      69$:
2938 014172 012637 001230      MOV      (SP)+,$TMP2      ;REPORT ERROR REGISTER
2939 014176 005726                      TST      (SP)+           ;NOT SET AS EXPECTED.
2940 014200 013737 177740 001232      MOV      @#LOADRS,$TMP3   ;RESET THE STACK.
2941 014206 013737 177742 001234      MOV      @#HIADRS,$TMP4
2942 014214 012737 001000 001236      MOV      #1000,$TMP5
2943 014222 012737 004420 001240      MOV      #4420,$TMP6
2944 014230 013737 177744 001242      MOV      @#MEMERR,$TMP7
2945
2946 014236 104131                      70$: ERROR 131
2947 014240 012737 177777 032344      MOV      #-1,$MANFL2     ;SIGNAL BAD REGISTER
2948 014246 012737 177777 032340      MOV      #-1,$MMRFL2
2949 014254 000705                      BR       65$
2950 014256 104416      MIDONE: RSET
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964 014260 000004
2965 014262 012737 000040 001274      TST26: SCOPE
2966 014266 000026                      MOV      #40,$TIMES     ;;DO 40 ITERATIONS
2967
2968 014270 012737 014624 032110      MJ-$TN-1
2969
2970 014276 113737 001102 001224      MOV      #TST27,$KAD     ;SET THE SKAD REGISTER
2971
2972 014304 104430                      MOV      #TST27,$KAD     ;IN CASE THE TEST ABORTS.
2973 014306 104432                      MOV      #TST27,$KAD
2974 014310 104434                      MOV      #TST27,$KAD
2975 014312 104436                      MOV      #TST27,$KAD
2976 014314 012737 014422 000114      MOV      #TST27,$KAD
2977 014322 012704 002000                      MOV      #TST27,$KAD
2978 014326 012702 177750                      MOV      #TST27,$KAD
2979 014332 012737 000044 177746      MOV      #TST27,$KAD
2980
2981
2982 014340 012705 014402                      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2983 014344 005715                      SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2984 014346 005715                      SKPBMM      ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
2985
2986
2987 014350 032737 000010 177752      SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2988 014356 001007                      MOV      #MJERRO,@#CACHVEC ;SET UP FOR THE ERROR.
                                MOV      #2000,R4           ;PATTERN TO BE PUT IN MAINT. REG.
                                MOV      #MAINT,R2
                                MOV      #S1M0,@#CONTRL ;FORCE SELECT GROUP 1 AND
                                ;FORCE MISS THE OTHER
                                ;GROUP
                                MOV      #MJ1,R5           ;MAKE MJ1 A HIT IN
                                TST      (R5)             ;GROUP GP.
                                TST      (R5)
                                BIT      #10,@#HITMIS     ;SEE IF REFERENCE ADDRESS
                                BNE      1$              ;IS A HIT.
    
```

CE  
CE

```

2989                                     ;IF NOT ERROR!
2990 014360 010537 001230                MOV    R5,$TMP2
2991 014364 012737 000001 001226        MOV    #1,$TMP1
2992 014372 104001                        ERROR  1
2993
2994 014374 104420                        SKIPT
2995                                     ;ERROR FATAL. GO TO NEXT TEST.
2996 014376 000240                        1$:   NOP
2997 014400 010412                        MOV    R4,(R2)
2998 014402 005012                        MJ1:  CLR    (R2)
2999                                     ;PUT THE PATTERN IN THE
3000                                     ;MAINTENANCE REGISTER.
3001                                     ;THE FETCH OF THIS NEXT
3002                                     ;INSTRUCTION SHOULD CAUSE
3003                                     ;A PARITY ERROR IN THE
3004                                     ;CACHE ADDRESS MEMORY GROUP GP.
3004 014404                                     MJ2:
3005 014404 010437 001230                MOV    R4,$TMP2
3006                                     ;REPORT ERROR. MAINTENANCE
3007 014410 104127                        1$:   ERROR  127
3008 014412 012737 177777 032344        MOV    #-1,MANFL2
3009 014420 000500                        BR     MJDONE
3010
3011 014422 022737 004440 177744        MJERR0: CMP   #4440,@MEMERR
3012 014430 001042                        BNE   69$
3013                                     ;DID THE ERROR REGISTER
3014 014432 022626                        64$:  CMP   (SP)+,(SP)+
3015 014434 005037 177572                        65$:  CLR   @MMR0
3016 014440 005037 172516                        CLR   @MMR3
3017 014444 012737 177777 177744        MOV    #-1,@MEMERR
3018 014452 005737 177744                        TST   @MEMERR
3019 014456 001416                        BEQ   68$
3020
3021 014460                                     66$:
3022 014460 013737 177740 001230        MOV    @LOADRS,$TMP2
3023 014466 013737 177742 001232        MOV    @HIADRS,$TMP3
3024 014474 013737 177744 001234        MOV    @MEMERR,$TMP4
3025
3026 014502 104130                        67$:  ERROR  130
3027 014504 012737 177777 032324        MOV    #-1,MMRFLG
3028 014512 000443                        BR     MJDONE
3029
3030 014514 022737 177740 177740        68$:  CMP   #177740,@LOADRS
3031 014522 001356                        BNE   66$
3032 014524 022737 000003 177742        CMP   #3,@HIADRS
3033 014532 001352                        BNE   66$
3034 014534 000432                        BR     MJDONE
3035
3036 014536                                     69$:
3037 014536 012637 001230                MOV    (SP)+,$TMP2
3038 014542 005726                        TST   (SP)+
3039 014544 013737 177740 001232        MOV    @LOADRS,$TMP3
3040 014552 013737 177742 001234        MOV    @HIADRS,$TMP4
3041 014560 012737 002000 001236        MOV    #2000,$TMP5
3042 014566 012737 004440 001240        MOV    #4440,$TMP6
3043 014574 013737 177744 001242        MOV    @MEMERR,$TMP7
3044
    
```

CE  
CE

```

3045 014602 104131 70$: ERROR 131
3046 014604 012737 177777 032344 MOV #-1,MANFL2 ;SIGNAL BAD REGISTER
3047 014612 012737 177777 032340 MOV #-1,MMRFL2
3048 014620 000705 BR 65$
3049 014622 104416 MJDONE: RSET
3050
3051
3052
3053 :*****
3054 :*TEST 27 CACHE MAINTENANCE AND ERROR REGISTERS TEST 13
3055 :*
3056 :*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3057 :*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE
3058 :*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
3059 :*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3060 :*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3061 :*TO THE CACHE.
3062 :*
3063 :*****
3064 014624 000004 TST27: SCOPE
3065 014626 012737 000040 001274 MOV #40,$TIMES ;;DO 40 ITERATIONS
3066 000027 MK-$TN-1
3067 014634 012737 015170 032110 MOV #TST30,SKAD ;SET THE SKAD REGISTER
3068 ;IN CASE THE TEST ABORTS.
3069 014642 113737 001102 001224 MOVB $TSTM,$TMP0
3070
3071 014650 104430 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3072 014652 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3073 014654 104434 SKPBMM ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
3074 014656 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3075 014660 012737 014766 000114 MOV #MKERRO,@#CACHVEC ;SET UP FOR THE ERROR.
3076 014666 012704 004000 MOV #4000,R4 ;PATTERN TO BE PUT IN MAINT. REG.
3077 014672 012702 177750 MOV #MAINT,R2
3078 014676 012737 000044 177746 MOV #S1M0,@#CONTRL ;FORCE SELECT GROUP 1 AND
3079 ;FORCE MISS THE OTHER
3080 ;GROUP
3081 014704 012705 014746 MOV #MK1,R5 ;MAKE MK1 A HIT IN
3082 014710 005715 TST (R5) ;GROUP GP.
3083 014712 005715 TST (R5)
3084
3085 ;SEE IF REFERENCE ADDRESS
3086 014714 032737 000010 177752 BIT #10,@#HITMIS ;IS A HIT.
3087 014722 001007 BNE 1$
3088 ;IF NOT ERROR.
3089 014724 010537 001230 MOV R5,$TMP2
3090 014730 012737 000001 001226 MOV #1,$TMP1
3091 014736 104001 ERROR 1
3092
3093 014740 104420 SKIPT ;ERROR FATAL. GO TO NEXT TEST.
3094
3095 014742 000240 1$: NOP ;PUT THE PATTERN IN THE
3096 014744 010412 MOV R4,(R2) ;MAINTENANCE REGISTER.
3097 014746 005012 MK1: CLR (R2) ;THE FETCH OF THIS NEXT
3098 ;INSTRUCTION SHOULD CAUSE
3099 ;A PARITY ERROR IN THE
3100 ;CACHE ADDRESS MEMORY GROUP GP.
    
```

```

3101
3102
3103 014750          MK2:          ;REPORT ERROR. MAINTENANCE
3104 014750 010437 001230      MOV      R4,$TMP2      ;FUNCTION FAILED TO
3105                                     ;CAUSE ERROR.
3106 014754 104127          1$:      ERROR 127
3107 014756 012737 177777 032344  MOV      #-1,MANFL2
3108 014764 000500          BR      MKDONE
3109
3110 014766 022737 004440 177744  MKERRO:  CMP      #4440,@MEMERR ;DID THE ERROR REGISTER
3111 014774 001042          BNE      69$           ;SET PROPERLY?
3112
3113 014776 022626          64$:      CMP      (SP)+,(SP)+ ;RESET THE STACK
3114 015000 005037 177572          65$:      CLR      @MMR0
3115 015004 005037 172516          CLR      @MMR3
3116 015010 012737 177777 177744  MOV      #-1,@MEMERR ;TRY TO CLEAR THE ERROR
3117 015016 005737 177744          TST      @MEMERR      ;REGISTER.
3118 015022 001416          BEQ      68$
3119
3120 015024          66$:          ;ERROR REGISTER WON'T
3121 015024 013737 177740 001230      MOV      @LOADRS,$TMP2 ;CLEAR
3122 015032 013737 177742 001232      MOV      @HIADRS,$TMP3
3123 015040 013737 177744 001234      MOV      @MEMERR,$TMP4
3124
3125 015046 104130          67$:      ERROR 130
3126 015050 012737 177777 032324  MOV      #-1,MMRFLG ;SIGNAL BAD REGISTER
3127 015056 000443          BR      MKDONE
3128
3129 015060 022737 177740 177740  68$:      CMP      #177740,@LOADRS ;SEE IF ADDRESS REGISTER
3130 015066 001356          BNE      66$           ;UNLOCKED.
3131 015070 022737 000003 177742      CMP      #3,@HIADRS
3132 015076 001352          BNE      66$
3133 015100 000432          BR      MKDONE
3134
3135 015102          69$:          ;REPORT ERROR REGISTER
3136 015102 012637 001230      MOV      (SP)+,$TMP2 ;NOT SET AS EXPECTED.
3137 015106 005726          TST      (SP)+      ;RESET THE STACK.
3138 015110 013737 177740 001232      MOV      @LOADRS,$TMP3
3139 015116 013737 177742 001234      MOV      @HIADRS,$TMP4
3140 015124 012737 004000 001236      MOV      #4000,$TMP5
3141 015132 012737 004440 001240      MOV      #4440,$TMP6
3142 015140 013737 177744 001242      MOV      @MEMERR,$TMP7
3143
3144 015146 104131          70$:      ERROR 131
3145 015150 012737 177777 032344  MOV      #-1,MANFL2 ;SIGNAL BAD REGISTER
3146 015156 012737 177777 032340  MOV      #-1,MMRFL2
3147 015164 000705          BR      65$
3148 015166 104416          MKDONE: RSET
3149
3150
3151
3152
3153
3154
3155
3156
    
```

```

*****
*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
    
```

```

3157 ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3158 ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3159 ;*TO THE CACHE.
3160 ;*
3161 ;*****
3162 015170 000004 TST30: SCOPE
3163 015172 012737 000040 001274 MOV #40,$TIMES ;;DO 40 ITERATIONS
3164 000030 ML-$TN-1
3165 ;
3166 015200 012737 015534 032110 MOV #TST31,SKAD ;SET THE SKAD REGISTER
3167 ;IN CASE THE TEST ABORTS
3168 015206 113737 001102 001224 MOVB $TSTM,$TMP0
3169 ;
3170 015214 104430 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3171 015216 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3172 015220 104434 SKPBMM ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
3173 015222 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3174 015224 012737 015332 000114 MOV #MLERR0,@CACHVEC ;SET UP FOR THE ERROR.
3175 015232 012704 000020 MOV #20,R4 ;PATTERN TO BE PUT IN MAINT. REG.
3176 015236 012702 177750 MOV #MAINT,R2
3177 015242 012737 000030 177746 MOV #SOM1,@CONTRL ;FORCE SELECT GROUP 0 AND
3178 ;FORCE MISS THE OTHER
3179 ;GROUP
3180 015250 012705 015312 MOV #ML1,R5 ;MAKE ML1 A HIT IN
3181 015254 005715 TST (R5) ;GROUP GP.
3182 015256 005715 TST (R5)
3183 ;
3184 ;SEE IF REFERENCE ADDRESS
3185 015260 032737 000010 177752 BIT #10,@HITMIS ;IS A HIT.
3186 015266 001007 BNE 1$
3187 ;IF NOT ERROR.
3188 015270 010537 001230 MOV R5,$TMP2
3189 015274 012737 000000 001226 MOV #0,$TMP1
3190 015302 104001 ERROR 1
3191 ;
3192 015304 104420 SKIPT ;ERROR FATAL. GO TO NEXT TEST.
3193 ;
3194 015306 000240 1$: NOP ;PUT THE PATTERN IN THE
3195 015310 010412 MOV R4,(R2) ;MAINTENANCE REGISTER.
3196 015312 005012 ML1: CLR (R2) ;THE FETCH OF THIS NEXT
3197 ;INSTRUCTION SHOULD CAUSE
3198 ;A PARITY ERROR IN THE
3199 ;CACHE DATA MEMORY GROUP GP.
3200 ;
3201 ;
3202 015314 ML2: ;REPORT ERROR. MAINTENANCE
3203 015314 010437 001230 MOV R4,$TMP2 ;FUNCTION FAILED TO
3204 ;CAUSE ERROR.
3205 015320 104127 1$: ERROR 127
3206 015322 012737 177777 032344 MOV #-1,MANF L2
3207 015330 000500 BP ML DONE
3208 ;
3209 015332 022737 004500 177744 MLERR0: CMP #4500,@MEMERR ;DID THE ERROR REGISTER
3210 015340 001042 BNE 69$ ;SET PROPFRLY?
3211 ;
3212 015342 022626 64$: (MP (SP)+,(SP)+ ;RESET THE STACK
    
```

```

3213 015344 005037 177572      65$: CLR @MMR0
3214 015350 005037 172516      CLR @MMR3
3215 015354 012737 177777 177744  MOV #-1,@MEMERR ;TRY TO CLEAR THE ERROR
3216 015362 005737 177744      TST @MEMERR ;REGISTER.
3217 015366 001416      BEQ 68$
3218
3219 015370      66$: MOV @LOADRS,$TMP2 ;ERROR REGISTER WON'T
3220 015370 013737 177740 001230      MOV @HIADRS,$TMP3 ;CLEAR
3221 015376 013737 177742 001232      MOV @MEMERR,$TMP4
3222 015404 013737 177744 001234
3223
3224 015412 104130      67$: ERROR 130
3225 015414 012737 177777 032324  MOV #-1,MMRFLG ;SIGNAL BAD REGISTER
3226 015422 000443      BR MLDONE
3227
3228 015424 022737 177740 177740 68$: CMP #177740,@LOADRS ;SEE IF ADDRESS REGISTER
3229 015432 001356      BNE 66$ ;UNLOCKED.
3230 015434 022737 000003 177742  CMP #3,@HIADRS
3231 015442 001352      BNE 66$
3232 015444 000432      BR MLDONE
3233
3234 015446      69$: MOV (SP)+,$TMP2 ;REPORT ERROR REGISTER
3235 015446 012637 001230      TST (SP)+ ;NOT SET AS EXPECTED.
3236 015452 005726      MOV @LOADRS,$TMP3 ;RESET THE STACK.
3237 015454 013737 177740 001232  MOV @HIADRS,$TMP4
3238 015462 013737 177742 001234  MOV #20,$TMP5
3239 015470 012737 000020 001236  MOV #4500,$TMP6
3240 015476 012737 004500 001240  MOV @MEMERR,$TMP7
3241 015504 013737 177744 001242
3242
3243 015512 104131      70$: ERROR 131
3244 015514 012737 177777 032344  MOV #-1,MANFL2 ;SIGNAL BAD REGISTER
3245 015522 012737 177777 032340  MOV #-1,MMRFL2
3246 015530 000705      BR 65$
3247 015532 104416      MLDONE: RSET
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261 015534 000004
3262 015536 012737 000040 001274  TST31: SCOPE
3263 000031      MN $TN-1
3264
3265 015544 012737 016100 032110  MOV #TST32,SKAD ;SET THE SKAD REGISTER
3266
3267 015552 113737 001102 001224  MOV $STNM,$TMP0 ;IN CASE THE TEST ABORT'S.
3268
    
```

```

:*****
:*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.
:*
:*****
    
```



3269	015560	104430			SKPBER				:IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3270	015562	104432			SKPBCN				:IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3271	015564	104434			SKPBMN				:IF THE MAINTENANCE REGISTER IS BAD SKIP THIS TEST.
3272	015566	104436			SKPBHM				:IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3273	015570	012737	015676	000114	MOV	#NMERRO,@#CACHVEC			:SET UP FOR THE ERROR.
3274	015576	012704	000040		MOV	#40,R4			:PATTERN TO BE PUT IN MAINT. REG.
3275	015602	012702	177750		MOV	#MAINT,R2			
3276	015606	012737	000030	177746	MOV	#SOM1,@#CONTRL			:FORCE SELECT GROUP 0 AND
3277									:FORCE MISS THE OTHER
3278									:GROUP
3279	015614	012705	015656		MOV	#NM1,R5			:MAKE NM1 A HIT IN
3280	015620	005715			TST	(R5)			:GROUP GP.
3281	015622	005715			TST	(R5)			
3282									
3283									:SEE IF REFERENCE ADDRESS
3284	015624	032737	000010	177752	BIT	#10,@#HITMIS			:IS A HIT.
3285	015632	001007			BNE	1\$			
3286									:IF NOT ERROR.
3287	015634	010537	001230		MOV	R5,\$TMP2			
3288	015640	012737	000000	001226	MOV	#0,\$TMP1			
3289	015646	104001			ERRCR	1			
3290									
3291	015650	104420			SKIPT				:ERROR FATAL. GO TO NEXT TEST.
3292									
3293	015652	000240			1\$: NOP				:PUT THE PATTERN IN THE
3294	015654	010412			MOV	R4,(R2)			:MAINTENANCE REGISTER.
3295	015656	005012			NM1: CLR	(R2)			:THE FETCH OF THIS NEXT
3296									:INSTRUCTION SHOULD CAUSE
3297									:A PARITY ERROR IN THE
3298									:CACHE DATA MEMORY GROUP GP.
3299									
3300									
3301	015660				NM2:				:REPORT ERROR. MAINTENANCE
3302	015660	010437	001230		MOV	R4,\$TMP2			:FUNCTION FAILED TO
3303									:CAUSE ERROR.
3304	015664	104127			1\$: ERROR	127			
3305	015666	012737	177777	032344	MOV	#-1,MANFL2			
3306	015674	000500			BR	NMDONE			
3307									
3308	015676	022737	004500	177744	NMERRO: CMP	#4500,@#MEMERR			:DID THE ERROR REGISTER
3309	015704	001042			BNE	69\$			:SET PROPERLY?
3310									
3311	015706	022626			64\$: CMP	(SP)+,(SP)+			:RESET THE STACK
3312	015710	005037	177572		65\$: CLR	@#MMR0			
3313	015714	005037	172516		CLR	@#MMR3			
3314	015720	012737	177777	177744	MOV	#-1,@#MEMERR			:TRY TO CLEAR THE ERROR
3315	015726	005737	177744		TST	@#MEMERR			:REGISTER.
3316	015732	001416			BEQ	68\$			
3317									
3318	015734				66\$: MOV	@#LOADRS,\$TMP2			:ERROR REGISTER WON'T
3319	015734	013737	177740	001230	MOV	@#HIADRS,\$TMP3			:CLEAR
3320	015742	013737	177742	001232	MOV	@#MEMERR,\$TMP4			
3321	015750	013737	177744	001234	MOV				
3322									
3323	015756	104130			67\$: ERROR	130			
3324	015760	012737	177777	032324	MOV	#-1,MMRFLG			:SIGNAL BAD REGISTER

```
3325 015766 000443 BR NMDONE
3326
3327 015770 022737 177740 177740 68$: CMP #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3328 015776 001356 BNE 66$ ;UNLOCKED.
3329 016000 022737 000003 177742 CMP #3,@#HIADRS
3330 016006 001352 BNE 66$
3331 016010 000432 BR NMDONE
3332
3333 016012 69$: ;REPORT ERROR REGISTER
3334 016012 012637 001230 MOV (SP)+,$TMP2 ;NOT SET AS EXPECTED.
3335 016016 005726 TST (SP)+ ;RESET THE STACK.
3336 016020 013737 177740 001232 MOV @#LOADRS,$TMP3
3337 016026 013737 177742 001234 MOV @#HIADRS,$TMP4
3338 016034 012737 000040 001236 MOV #40,$TMP5
3339 016042 012737 004500 001240 MOV #4500,$TMP6
3340 016050 013737 177744 001242 MOV @#MEMERR,$TMP7
3341
3342 016056 104131 70$: ERROR 131
3343 016060 012737 177777 032344 MOV #-1,$MANFL2 ;SIGNAL BAD REGISTER
3344 016066 012737 177777 032340 MOV #-1,$MMRFL2
3345 016074 000705 BR 65$
3346 016076 104416 NMDONE: RSET
3347
3348
3349
3350 ;*****
3351 ;*TEST 32 CACHE MAINTENANCE AND ERROR REGISTERS TEST 16
3352 ;*
3353 ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3354 ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
3355 ;*LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3356 ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3357 ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3358 ;*TO THE CACHE.
3359 ;*
3360 ;*****
3361 TST32: SCOPE
3362 MOV #40,$TIMES ;:DO 40 ITERATIONS
3363 MO=$TN-1
3364 MOV #TST33,$SKAD ;:SET THE SKAD REGISTER
3365 ;:IN CASE THE TEST ABORTS.
3366 MOVB $TSTNM,$TMP0
3367
3368 SKPBER ;:IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3369 SKPBCN ;:IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3370 SKPBMM ;:IF THE MAINTENANCE REGISTER IS BAD SKIP THIS TEST.
3371 SKPBHM ;:IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3372 MOV #MOERRO,@#CACHVEC ;:SET UP FOR THE ERROR.
3373 MOV #100,$R4 ;:PATTERN TO BE PUT IN MAINT. REG.
3374 MOV #MAINT,$R2
3375 MOV #S1M0,@#CONTRL ;:FORCE SELECT GROUP 1 AND
3376 ;:FORCE MISS THE OTHER
3377 ;:GROUP
3378 MOV #M01,$R5 ;:MAKE M01 A HIT IN
3379 TST ($R5) ;:GROUP GP.
3380 TST ($R5)
```

```

3381
3382
3383 016170 032737 000010 177752 BIT #10,@WHITMIS ;SEE IF REFERENCE ADDRESS
3384 016176 001007 BNE 1$ ;IS A HIT.
3385 ;IF NOT ERROR!
3386 016200 010537 001230 MOV R5,$TMP2
3387 016204 012737 000001 001226 MOV #1,$TMP1
3388 016212 104001 ERROR 1
3389
3390 016214 104420 SKIPT ;ERROR FATAL. GO TO NEXT TEST.
3391
3392 016216 000240 1$: NOP ;PUT THE PATTERN IN THE
3393 016220 010412 MOV R4,(R2) ;MAINTENANCE REGISTER.
3394 016222 005012 MO1: CLR (R2) ;THE FETCH OF THIS NEXT
3395 ;INSTRUCTION SHOULD CAUSE
3396 ;A PARITY ERROR IN THE
3397 ;CACHE DATA MEMORY GROUP GP.
3398
3399
3400 016224 MO2: ;REPORT ERROR. MAINTENANCE
3401 016224 010437 001230 MOV R4,$TMP2 ;FUNCTION FAILED TO
3402 ;CAUSE ERROR.
3403 016230 104127 1$: ERROR 127
3404 016232 012737 177777 032344 MOV #-1,MANFL2
3405 016240 000500 BR MODONE
3406
3407 016242 022737 004600 177744 MOERRO: CMP #4600,@MEMERR ;DID THE ERROR REGISTER
3408 016250 001042 BNE 69$ ;SET PROPERLY?
3409
3410 016252 022626 64$: CMP (SP)+,(SP)+ ;RESET THE STACK
3411 016254 005037 177572 65$: CLR @MMR0
3412 016260 005037 172516 CLR @MMR3
3413 016264 012737 177777 177744 MOV #-1,@MEMERR ;TRY TO CLEAR THE ERROR
3414 016272 005737 177744 TST @MEMERR ;REGISTER.
3415 016276 001416 BEQ 68$
3416
3417 016300 66$: ;ERROR REGISTER WON'T
3418 016300 013737 177740 001230 MOV @LOADRS,$TMP2 ;CLEAR
3419 016306 013737 177742 001232 MOV @HIADRS,$TMP3
3420 016314 013737 177744 001234 MOV @MEMERR,$TMP4
3421
3422 016322 104130 67$: ERROR 130
3423 016324 012737 177777 032324 MOV #-1,MMRFLG ;SIGNAL BAD REGISTER
3424 016332 000443 BR MODONE
3425
3426 016334 022737 177740 177740 68$: CMP #177740,@LOADRS ;SEE IF ADDRESS REGISTER
3427 016342 001356 BNE 66$ ;UNLOCKED.
3428 016344 022737 000003 177742 CMP #3,@HIADRS
3429 016352 001352 BNE 66$
3430 016354 000432 BR MODONE
3431
3432 016356 69$: ;REPORT ERROR REGISTER
3433 016356 012637 001230 MOV (SP)+,$TMP2 ;NOT SET AS EXPECTED.
3434 016362 005726 TST (SP)+ ;RESET THE STACK.
3435 016364 013737 177740 001232 MOV @LOADRS,$TMP3
3436 016372 013737 177742 001234 MOV @HIADRS,$TMP4
    
```

```

3437 016400 012737 000100 001236      MOV      #100,$TMP5
3438 016406 012737 004600 001240      MOV      #4600,$TMP6
3439 016414 013737 177744 001242      MOV      @MEMERR,$TMP7
3440
3441 016422 104131      70$:     ERROR      131
3442 016424 012737 177777 032344      MOV      #-1,MANFL2      ;SIGNAL BAD REGISTER
3443 016432 012737 177777 032340      MCV      #-1,MMRFL2
3444 016440 000705      BR        65$
3445 016442 104416      MODONE:  RSET
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459 016444 000004      *****
3460 016446 012737 000040 001274      *TEST 33      CACHE MAINTENANCE AND ERROR REGISTERS TEST 17
3461      000033      *
3462
3463 016454 012737 017010 032110      *THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3464
3465 016462 113737 001102 001224      *TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
3466
3467 016470 104430      *HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3468 016472 104432      *ABILITY TO SET CORRECTLY FOR THIS ERROR.
3469 016474 104434      *THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3470 016476 104436      *TO THE CACHE.
3471 016500 012737 016606 000114      *
3472 016506 012704 000200      *****
3473 016512 012702 177750      TST33:  SCOPE
3474 016516 012737 000044 177746      MOV      #40,$TIMES      ;;DO 40 ITERATIONS
3475
3476
3477 016524 012705 016566      MP-$TN-1
3478 016530 005715      MOV      #TST34,$KAD      ;SET THE SKAD REGISTER
3479 016532 005715      MOV      $TSTNM,$TMP0      ;IN CASE THE TEST ABORTS.
3480
3481
3482 016534 032737 000010 177752      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3483 016542 001007      SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3484
3485 016544 010537 001230      SKPBMM      ;IF THE MAINTENANCE REGISTER IS BAD SKIP THIS TEST.
3486 016550 012737 000001 001226      SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3487 016556 104001      MOV      #MPERRO,@CACHVEC      ;SET UP FOR THE ERROR.
3488
3489 016560 104420      MOV      #200,R4      ;PATTERN TO BE PUT IN MAINT. REG.
3490
3491 016562 000240      MOV      #MAINT,R2
3492 016564 010412      MOV      #S1MO,@CONTRL      ;FORCE SELECT GROUP 1 AND
                                ;FORCE MISS THE OTHER
                                ;GROUP
                                ;MAKE MP1 A HIT IN
                                ;GROUP GP.
                                ;SEE IF REFERENCE ADDRESS
                                ;IS A HIT.
                                ;IF NOT ERROR.
                                ;ERROR FATAL. GO TO NEXT TEST.
                                ;PUT THE PATTERN IN THE
                                ;MAINTENANCE REGISTER.
                                1$:
                                MOV      R4,(R2)
    
```

```

3493 016566 005012 MP1: CLR (R2) ;THE FETCH OF THIS NEXT
3494 ;INSTRUCTION SHOULD CAUSE
3495 ;A PARITY ERROR IN THE
3496 ;CACHE DATA MEMORY GROUP GP.
3497
3498
3499 016570 MP2: ;REPORT ERROR. MAINTENANCE
3500 016570 010437 001230 MOV R4,$TMP2 ;FUNCTION FAILED TO
3501 ;CAUSE ERROR.
3502 016574 104127 1$: ERROR 127
3503 016576 012737 177777 032344 MOV #-1,MANFL2
3504 016604 000500 BR MPDONE
3505
3506 016606 022737 004600 177744 MPPER0: CMP #4600,@MEMERR ;DID THE ERROR REGISTER
3507 016614 001042 BNE 69$ ;SET PROPERLY?
3508
3509 016616 022626 64$: CMP (SP)+,(SP)+ ;RESET THE STACK
3510 016620 005037 177572 65$: CLR @MMR0
3511 016624 005037 172516 CLR @MMR3
3512 016630 012737 177777 177744 MOV #-1,@MEMERR ;TRY TO CLEAR THE ERROR
3513 016636 005737 177744 TST @MEMERR ;REGISTER.
3514 016642 001416 BEQ 68$
3515
3516 016644 66$: ;ERROR REGISTER WON'T
3517 016644 013737 177740 001230 MOV @LOADRS,$TMP2 ;CLEAR
3518 016652 013737 177742 001232 MOV @HIADRS,$TMP3
3519 016660 013737 177744 001234 MOV @MEMERR,$TMP4
3520
3521 016666 104130 67$: ERROR 130
3522 016670 012737 177777 032324 MOV #-1,MMRFLG ;SIGNAL BAD REGISTER
3523 016676 000443 BR MPDONE
3524
3525 016700 022737 177740 177740 68$: CMP #177740,@LOADRS ;SEE IF ADDRESS REGISTER
3526 016706 001356 BNE 66$ ;UNLOCKED.
3527 016710 022737 000003 177742 CMP #3,@HIADRS
3528 016716 001352 BNE 66$
3529 016720 000432 BR MPDONE
3530
3531 016722 69$: ;REPORT ERROR REGISTER
3532 016722 012637 001230 MOV (SP)+,$TMP2 ;NOT SET AS EXPECTED.
3533 016726 005726 TST (SP)+ ;RESET THE STACK.
3534 016730 013737 177740 001232 MOV @LOADRS,$TMP3
3535 016736 013737 177742 001234 MOV @HIADRS,$TMP4
3536 016744 012737 000200 001236 MOV #200,$TMP5
3537 016752 012737 004600 001240 MOV #4600,$TMP6
3538 016760 013737 177744 001242 MOV @MEMERR,$TMP7
3539
3540 016766 104131 70$: ERROR 131
3541 016770 012737 177777 032344 MOV #-1,MANFL2 ;SIGNAL BAD REGISTER
3542 016776 012737 177777 032340 MOV #-1,MMRFL2
3543 017004 000705 BR 65$
3544 017006 104416 MPDONE: RSET
3545
3546
3547
3548
    
```

```

3549
3550
3551 :*****
3552 :*TEST 34      CACHE MAINTENANCE AND ERROR REGISTERS TEST 20
3553 :*
3554 :*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3555 :*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3556 :*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3557 :*THE MAINTENANCE REGISTER IS USED TO MAKE THAT REFERENCE CAUSE A
3558 :*MAIN MEMORY ADDRESS AND CONTROL LINES PARITY ERROR ON THE
3559 :*MAIN MEMORY BUS.
3560 :*
3561 :*****
3561 017010 000004 TST34: SCOPE
3562 017012 012737 000040 001274      MOV      #40,$TIMES      ;;DO 40 ITERATIONS
3563      000034      MR=$TN-1
3564
3565 017020 012737 017440 032110      MOV      #TST35,SKAD    ;SET THE SKAD REGISTER
3566      ;IN CASE THE TEST ABORTS.
3567 017026 113737 001102 001224      MOVB     $TSTNM,$TMP0
3568
3569 017034 104430      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3570 017036 104432      SKPBCN     ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3571 017040 104434      SKPBMM     ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
3572 017042 104436      SKPBHM     ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3573 017044 104422      MMSKIP
3574 017046 012737 017230 000114      MOV      #MRERRO,@#CACHVEC ;SET UP FOR THE ERROR.
3575 017054 012737 031736 000004      MOV      #CPSPUR,@#ERRVEC  ;NOTE THAT WHEN THIS ERROR
3576      ;ON THE MAIN MEMORY ADDRESS
3577      ;AND CONTROL LINES OCCURS
3578      ;A TIME OUT WILL RESULT ON THE
3579      ;UNIBUS!! THIS WILL CAUSE A
3580      ;TRAP TO VECTOR ERRVEC BEFORE
3581      ;THE TRAP TO CACHVEC OCCURS! BOTH
3582      ;WILL OCCUR!
3583 017062 012746 177777      MOV      #-1,-(SP)      ;PUT A MARKER ON THE STACK
3584
3585 017066 012700 172340      MOV      #KIPAR0,R0     ;SET UP MEMORY MANAGEMENT
3586      ;TO RELOCATE EVERYTHING
3587 017072 012702 172300      MOV      #KIPDR0,R2     ;THROUGH THE UNIBUS
3588 017076 012703 000007      MOV      #7,R3          ;MAP PASSIVELY TO MEMORY,
3589 017102 005004      CLR      R4             ;BY PASSIVELY IS MEANT
3590 017104 012705 170200      MOV      #MAPL00,R5     ;THAT ADDRESS ARE
3591      ;RELOCATED TO THEMSELVES.
3592 017110 012722 077406      64$: MOV      #77406,(R2)+
3593 017114 010401      MOV      R4,R1
3594 017116 072127 000006      ASH     #6,R1
3595 017122 010125      MOV      R1,(R5)+
3596 017124 005025      CLR     (R5)+
3597 017126 010410      MOV      R4,(R0)
3598 017130 062720 170000      ADD     #170000,(R0)+
3599 017134 062704 000200      ADD     #200,R4
3600 017140 077315      SOB     R3,64$
3601 017142 012710 177600      MOV      #177600,(R0)
3602 017146 012712 077406      MOV      #77406,(R2)
3603
3604 017152 012737 000060 172516      MOV      #60,@#MMR3     ;TURN ON THE MAPPING BO; AND
    
```

```

3605 017160 012737 000001 177572      MOV      #1,@MMR0      ;ENABLE 22 BIT MODE ADDRESSING.
3606
3607 017166 012737 000014 177746      MOV      #MOM1,@CONTRL ;FORCE MISSES TO BOTH GROUPS.
3608 017174 012702 177750      MOV      #MAINT,R2
3609 017200 000240      NOP
3610 017202 012712 000002      MOV      #2,(R2)      ;FOR SCOPING WITH AN OSCILLOSCOPE
3611                                     ;SET UP THE FORCE ERROR BIT IN
3612 017206 005012      CLR      (R2)         ;THE MAINTENANCE REGISTER.
3613                                     ;THE FETCH OF THIS INSTRUCTION
3614                                     ;SHOULD RESULT IN A PARITY ERROR
3615                                     ;ON THE MAIN MEMORY ADDRESS AND CONTROL
3616                                     ;LINES. BECAUSE THIS REFERENCE
3617                                     ;IS BEING MADE OVER THE UNIBUS
3618                                     ;A UNIBUS TIME OUT WILL OCCUR
3619                                     ;RESULTING IN AN ABORT TO VECTOR
3620                                     ;ERRVEC. THEN IMMEDIATELY FOLLOWING
3621                                     ;THIS ABORT TO ERRVEC, THE
3622                                     ;PARITY ERROR WILL CAUSE A TRAP
3623                                     ;TO CACHVEC!!!
3624 017210      MR1:
3625 017210 012737 000002 001230      MOV      #2,$TMP2     ;REPORT FAILURE OF THE MAINTENANCE
3626 017216 104127      1$:      ERROR          ;TO FORCE THE ERROR.
3627 017220 012737 177777 032344      MOV      #-1,MANFL2
3628 017226 000503      BR
3629
3630 017230 022766 177777 000010      MRERR0: CMP      #-1,10(SP)  ;DID 2 TRAPS OCCUR? SEE WHERE
3631                                     ;THE MARKER IS ON THE STACK!
3632 017236 001401      BEQ
3633 017240 104000      ERROR
3634
3635 017242 022737 002402 177744      MR2:      CMP      #2402,@MEMERR
3636 017250 001430      BEQ      MR3         ;DID THE ERROR REGISTER GET
3637                                     ;SET CORRECTLY.
3638                                     ;IF NOT REPORT THE ERROR.
3639 017252 022626      CMP      (SP)+,(SP)+
3640 017254 012637 001230      MOV      (SP)+,$TMP2
3641 017260 022626      CMP      (SP)+,(SP)+
3642 017262 013737 177740 001232      MOV      @LOADRS,$TMP3
3643 017270 013737 177742 001234      MOV      @HIADRS,$TMP4
3644 017276 012737 000002 001236      MOV      #2,$TMP5
3645 017304 012737 002402 001240      MOV      #2402,$TMP6
3646 017312 013737 177744 001242      MOV      @MEMERR,$TMP7
3647 017320 104131      1$:      ERROR          131
3648 017322 012737 177777 032344      MOV      #-1,MANFL2
3649 017330 000402      BR      MR4
3650
3651 017332 062706 000012      MR3:      ADD      #12,SP      ;RESET THE STACK.
3652
3653 017336 005037 177572      MR4:      CLR      @MMR0
3654 017342 005037 172516      CLR      @MMR3
3655 017346 012737 177777 177744      MOV      #-1,@MEMERR  ;TRY TO CLR THE ERROR REG.
3656 017354 005737 177744      TST      @MEMERR
3657 017360 001416      BEQ      MR6
3658
3659 017362      MR5:
3660 017362 013737 177740 001230      MOV      @LOADRS,$TMP2 ;THE ERROR REGISTER WON'T CLR.
    
```

```

3661 017370 013737 177742 001232      MOV      @#HIADRS,$TMP3
3662 017376 013737 177744 001234      MOV      @#MEMERR,$TMP4
3663 017404 104130          1$:      ERROR    130
3664 017406 012737 177777 032324      MOV      #-1,$MRFLG
3665 017414 000410          BR        MRDONE
3666
3667 017416 022737 177740 177740      MR6:     CMP      #177740,@#LOADRS      ;SEE IF THE ADDRESS REGISTER
3668 017424 001356          BNE      MR5                ;GOT RESET.
3669 017426 022737 000003 177742      CMP      #3,@#HIADRS
3670 017434 001352          BNE      MR5
3671
3672 017436 104416          MRDONE: RSET
3673
3674          :*****
3675          :*TEST 35      CACHE MAINTENANCE AND ERROR REGISTERS TEST 21
3676          :*
3677          :*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3678          :*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3679          :*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3680          :*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
3681          :*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN EVEN WORD IN THE
3682          :*PAIR, WHICH IS ALSO THE WANTED WORD.
3683          :*
3684          :*****
3685 017440 000004          TST35:  SCOPE
3686 017442 012737 000040 001274      MOV      #40,$TIMES      ;;DO 40 ITERATIONS
3687          MS=$TN-1
3688
3689 017450 012737 020060 032110      MOV      #TST36,$SKAD      ;SET THE SKAD REGISTER
3690          ;IN CASE THE TEST ABORTS.
3691 017456 113737 001102 001224      MOV      $TSTNM,$TMP0
3692
3693 017464 104430          SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3694 017466 104432          SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3695 017470 104434          SKPBMM      ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3696 017472 104436          SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3697 017474 104422          MMSKIP
3698 017476 012737 017656 000114      MOV      #MSERRO,@#CACHVEC      ;SET UP FOR THE ERROR
3699
3700 017504 012700 172340          MOV      #KIPARO,R0      ;SET UP MEMORY MANAGEMENT
3701          ;TO RELOCATE EVERYTHING
3702 017510 012702 172300          MOV      #KIPDR0,R2      ;THROUGH THE UNIBUS
3703 017514 012703 000007          MOV      #7,R3          ;MAP PASSIVELY TO MEMORY,
3704 017520 005004          CLR      R4            ;BY PASSIVELY IS MEANT
3705 017522 012705 170200          MOV      #MAPL00,R5      ;THAT ADDRESS ARE
3706          ;RELOCATED TO THEMSELVES.
3707 017526 012722 077406          64$:     MOV      #77406,(R2)+
3708 017532 010401          MOV      R4,R1
3709 017534 072127 000006          ASH      #6,R1
3710 017540 010125          MOV      R1,(R5)+
3711 017542 005025          CLR      (R5)+
3712 017544 010410          MOV      R4,(R0)
3713 017546 062720 170000          ADD      #170000,(R0)+
3714 017552 062704 000200          ADD      #200,R4
3715 017556 077315          SOB      R3,64$
3716 017560 012710 177600          MOV      #177600,(R0)
    
```



```

3717 017564 012712 077406      MOV      #77406,(R2)
3718
3719 017570 012737 000060 172516      MOV      #60,@MMR3      ;TURN THE MAP AND ENABLE
3720 017576 012737 000001 177572      MOV      #1,@MMR0      ;22 BIT MODE ADDRESSING.
3721 017604 012704 010000      MOV      #10000,R4     ;PATTERN FOR THE MAINTENANCE
3722 017610 012702 177750      MOV      #MAINT,R2     ;REGISTER.
3723 017614 012737 000014 177746      MOV      #M1MO,@CONTRL ;FORCE MISSES TO BOTH GROUPS.
3724 017622 000402
3725
3726      017624      LOC=.      ;GET THE PC TO AN EVEN WORD BOUNDARY
3727      017624      LOC=-4&LOC
3728      017630      LOC=LOC+4
3729      017630      .=LOC
3730
3731 017630 000240      MS1:     NOP
3732 017632 010412      MOV      R4,(R2)      ;TURN ON THE MAINTENANCE REGISTER.
3733 017634 005701      MS2:     TST      R1
3734 017636 005012      CLR      (R2)
3735
3736 017640      MS3:
3737 017640 010437 001230      MOV      R4,$TMP2     ;REPORT ERROR. MAINTENANCE
3738      ;FUNCTION FAILED TO
3739      ;CAUSE ERROR.
3739 017644 104127      1$:     ERROR 127
3740 017646 012737 177777 032344      MOV      #-1,MANFL2
3741 017654 000500      BR      MSDONE
3742
3743 017656 022737 023404 177744      MSERRO: CMP      #23404,@MEMERR ;DID THE ERROR REGISTER
3744 017664 001042      BNE      69$          ;SET PROPERLY?
3745
3746 017666 022626      64$:     CMP      (SP)+,(SP)+ ;RESET THE STACK
3747 017670 005037 177572      65$:     CLR      @MMR0
3748 017674 005037 172516      CLR      @MMR3
3749 017700 012737 177777 177744      MOV      #-1,@MEMERR ;TRY TO CLEAR THE ERROR
3750 017706 005737 177744      TST      @MEMERR     ;REGISTER.
3751 017712 001416      BEQ     68$
3752
3753 017714      66$:
3754 017714 013737 177740 001230      MOV      @LOADRS,$TMP2 ;ERROR REGISTER WON'T
3755 017722 013737 177742 001232      MOV      @HIADRS,$TMP3 ;CLEAR
3756 017730 013737 177744 001234      MOV      @MEMERR,$TMP4
3757
3758 017736 104130      67$:     ERROR 130
3759 017740 012737 177777 032324      MOV      #-1,MMRFLG   ;SIGNAL BAD REGISTER
3760 017746 000443      BR      MSDONE
3761
3762 017750 022737 177740 177740      68$:     CMP      #177740,@LOADRS ;SEE IF ADDRESS REGISTER
3763 017756 001356      BNE      66$          ;UNLOCKED.
3764 017760 022737 000003 177742      CMP      #3,@HIADRS
3765 017766 001352      BNE      66$
3766 017770 000432      BR      MSDONE
3767
3768 017772      69$:
3769 017772 012637 001230      MOV      (SP)+,$TMP2 ;REPORT ERROR REGISTER
3770 017776 005726      TST      (SP)+      ;NOT SET AS EXPECTED.
3771 020000 013737 177740 001232      MOV      @LOADRS,$TMP3 ;RESET THE STACK.
3772 020006 013737 177742 001234      MOV      @HIADRS,$TMP4
    
```

```

3773 020014 012737 010000 001236      MOV      #10000,$TMP5
3774 020022 012737 023404 001240      MOV      #23404,$TMP6
3775 020030 015737 177744 001242      MOV      @MEMERR,$TMP7
3776
3777 020036 104131          70$:  ERROR  131
3778 020040 012737 177777 032344      MOV      #-1,$MANFL2      ;SIGNAL BAD REGISTER
3779 020046 012737 177777 032340      MOV      #-1,$MMRFL2
3780 020054 000705          BR      65$
3781 020056 104416      MSDONE: RSET
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794 020060 000004          ;*****
3795 020062 012737 000040 001274      ;*TEST 36      CACHE MAINTENANCE AND ERROR REGISTERS TEST 22
3796          000036          ;*
3797          ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3798          ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3799          ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3800          ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
3801          ;*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN ODD WORD IN THE
3802          ;*PAIR, WHICH IS ALSO THE WANTED WORD.
3803          ;*
3804          ;*****
3805          TST36:  SCOPE
3806          MOV      #40,$TIMES      ;;DO 40 ITERATIONS
3807          MT=$TN-1
3808          MOV      #TST3',SKAD      ;SET THE SKAD REGISTER
3809          ;IN CASE THE TEST ABORTS.
3810          MOV      $TSTNM,$TMP0
3811          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3812          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3813          SKPBMM          ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
3814          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3815          MMSKIP
3816
3817 020116 012700 172340      MOV      #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
3818          ;TO RELOCATE EVERYTHING
3819          MOV      #KIPDR0,R2      ;THROUGH THE UNIBUS
3820          MOV      #7,R3          ;MAP PASSIVELY TO MEMORY,
3821          CLR      R4            ;BY PASSIVELY IS MEANT
3822          MOV      #MAPL00,R5      ;THAT ADDRESS ARE
3823          ;RELOCATED TO THEMSELVES.
3824
3825 020140 012722 077406      64$:  MOV      #77406,(R2)+
3826          MOV      R4,R1
3827          ASH      #6,R1
3828          MOV      R1,(R5)+
3829          CLR      (R5)+
3830          MOV      R4,(R0)
3831          ADD      #170000,(R0)+
3832          ADD      #200,R4
3833          SOB      R3,64$
3834          MOV      #177600,(R0)
3835          MOV      #77406,(R2)
3836
3837 020202 012737 000060 172516      MOV      #60,@MMR3      ;TURN ON THE MAP AND 22-BIT
3838 020210 012737 000001 177572      MOV      #1,@MMR0      ;MODE ADDRESSING.
    
```

```

3829 020216 012737 020302 000114      MOV      @MTERRO,@CACHVEC      ;SET UP FOR THE ERROR.
3830 020224 012737 000014 177746      MOV      @MOM1,@CONTRL        ;FORCE MISSES TO BOTH GROUPS.
3831 020232 012704 040000                MOV      #40000,R4            ;PATTERN TO BE PUT IN MAINT.
3832 020236 012702 177750                MOV      @MAINT,R2            ;REG.
3833 020242 000403                BR       MT1
3834
3835                020244                LOC=      ;GET THE PC TO AN EVEN WORD BOUNDARY.
3836                020244                LOC=-4&LOC
3837                020250                LOC=LOC+4
3838                020250                .=LOC
3839
3840 020250 000240                NOP
3841 020252 000240      MT1:      NOP      ;NOP FOR SCOPING WITH AN OSCILLOSCOPE.
3842 020254 010412      MOV      R4,(R2)      ;SET THE MAINT. REG.
3843 020256 005701      TST      R1            ;THE REFERENCE TO THIS INSTRUCTION SHOULD CAUSE A PARITY
3844 020260 005012      CLR      (R2)          ;ABORT CAUSED BY DETECTION OF BAD PARITY ON
3845 020262 000240      NOP      ;THE WANTED, ODD, WORD IN THIS PAIR.
3846
3847
3848 020264                MT2:
3849 020264 010437 001230      MOV      R4,$TMP2        ;REPORT ERROR. MAINTENANCE
3850                ;FUNCTION FAILED TO
3851                ;CAUSE ERROR.
3851 020270 104127      1$:      ERROR 127
3852 020272 012737 177777 032344      MOV      #-1,MANFL2
3853 020300 000500      BR       MTDONE
3854
3855 020302 022737 023410 177744      MTERRO:  CMP      #23410,@MEMERR ;DID THE ERROR REGISTER
3856 020310 001042      BNE      69$            ;SET PROPERLY?
3857
3858 020312 022626      64$:      CMP      (SP)+,(SP)+      ;RESET THE STACK
3859 020314 005037 177572      65$:      CLR      @MMR0
3860 020320 005037 172516      CLR      @MMR3
3861 020324 012737 177777 177744      MOV      #-1,@MEMERR      ;TRY TO CLEAR THE ERROR
3862 020332 005737 177744      TST      @MEMERR          ;REGISTER.
3863 020336 001416      BEQ      68$
3864
3865 020340                66$:
3866 020340 013737 177740 001230      MOV      @LOADRS,$TMP2    ;ERROR REGISTER WON'T
3867 020346 013737 177742 001232      MOV      @HIADRS,$TMP3    ;CLEAR
3868 020354 013737 177744 001234      MOV      @MEMERR,$TMP4
3869
3870 020362 104130      67$:      ERROR 130
3871 020364 012737 177777 032324      MOV      #-1,MMRFLG      ;SIGNAL BAD REGISTER
3872 020372 000443      BR       MTDONE
3873
3874 020374 022737 177740 177740      68$:      CMP      #177740,@LOADRS ;SEE IF ADDRESS REGISTER
3875 020402 001356      BNE      66$            ;UNLOCKED.
3876 020404 022737 000003 177742      CMP      #3,@HIADRS
3877 020412 001352      BNE      66$
3878 020414 000432      BR       MTDONE
3879
3880 020416                69$:
3881 020416 012637 001230      MOV      (SP)+,$TMP2      ;REPORT ERROR REGISTER
3882 020422 005726      TST      (SP)+            ;NOT SET AS EXPECTED.
3883 020424 013737 177740 001232      MOV      @LOADRS,$TMP3    ;RESET THE STACK.
3884 020432 013737 177742 001234      MOV      @HIADRS,$TMP4
    
```

```

3885 020440 012737 040000 001236      MOV      #40000,$TMP5
3886 020446 012737 023410 001240      MOV      #23410,$TMP6
3887 020454 013737 177744 001242      MOV      @MEMERR,$TMP7
3888
3889 020462 104131          70$:    ERROR 131
3890 020464 012737 177777 032344      MOV      #-1,MANFL2      ;SIGNAL BAD REGISTER
3891 020472 012737 177777 032340      MOV      #-1,MMRFL2
3892 020500 000705          BR      65$
3893 020502 104416          MTDONE: RSET
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906 020504 000004          TST37:  SCOPE
3907 020506 012737 000040 001274      MOV      #40,$TIMES      ;;DO 40 ITERATIONS
3908 000037          MU $TN-1
3909
3910 020514 012737 021124 032110      MOV      #TST40,SKAD      ;SET THE SKAD REGISTER
3911                                     ;IN CASE THE TEST ABORTS.
3912 020522 113737 001102 001224      MOV      $TSTNM,$TMP0
3913
3914 020530 104430          SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3915 020532 104432          SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3916 020534 104434          SKPBMM      ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3917 020536 104436          SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3918 020540 104422          MMSKIP
3919
3920 020542 012700 172340      MOV      #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
3921                                     ;TO RELOCATE EVERYTHING
3922 020546 012702 172300      MOV      #KIPDR0,R2      ;THROUGH THE UNIBUS
3923 020552 012703 000007      MOV      #7,R3           ;MAP PASSIVELY TO MEMORY,
3924 020556 005004          CLR      R4             ;BY PASSIVELY IS MEANT
3925 020560 012705 170200      MOV      #MAPL00,R5      ;THAT ADDRESS ARE
3926                                     ;RELOCATED TO THEMSE VES.
3927 020564 012722 077406          64$:    MOV      #77406,(R2)+
3928 020570 010401          MOV      R4,R1
3929 020572 072127 000006          ASH      #6,R1
3930 020576 010125          MOV      R1,(R5)+
3931 020600 005025          CLR      (R5)+
3932 020602 010410          MOV      R4,(R0)
3933 020604 062720 170000          ADD      #170000,(R0)+
3934 020610 062704 000200          ADD      #200,R4
3935 020614 077315          SOB      R3,64$
3936 020616 012710 177600          MOV      #177600,(R0)
3937 020622 012712 077406          MOV      #77406,(R2)
3938
3939 020626 012737 000060 172516      MOV      #60,@MMR3      ;TURN ON THE MAP AND
3940 020634 012737 000001 177572      MOV      #1,@MMR0      ;22-BIT MODE ADDRESSING
    
```

```

*****
*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 / 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 .
*****
    
```

```

*****
TST37:  SCOPE
MOV      #40,$TIMES      ;;DO 40 ITERATIONS
MU $TN-1
MOV      #TST40,SKAD      ;SET THE SKAD REGISTER
;IN CASE THE TEST ABORTS.
MOV      $TSTNM,$TMP0
;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
    
```

```

64$:    MOV      #77406,(R2)+
MOV      R4,R1
ASH      #6,R1
MOV      R1,(R5)+
CLR      (R5)+
MOV      R4,(R0)
ADD      #170000,(R0)+
ADD      #200,R4
SOB      R3,64$
MOV      #177600,(R0)
MOV      #77406,(R2)
MOV      #60,@MMR3      ;TURN ON THE MAP AND
MOV      #1,@MMR0      ;22-BIT MODE ADDRESSING
    
```

```

3941 020642 012737 020722 000114      MOV    #MUERR0,@#CACHVEC      ;SETUP FOR THE ERROR.
3942 020650 012737 000030 177746      MOV    #SOM1,@#CONTRL        ;SELECT GROUP ADDRESS
3943 020656 012704 000400              MOV    #400,R4               ;PATTERN TO BE LOADED IN THE
3944 020662 012702 177750              MOV    #MAINT,R2             ;MAINTENANCE REG.
3945 020666 000403      BR     MU1
3946
3947          020670      LOC=.                          ;GET THE PC TO AN EVEN WORD BOUNDARY.
3948          020670      LOC=-4&LOC
3949          020674      LOC=LOC+4
3950          020674      .=LOC
3951
3952 020674 000240      NOP
3953 020676 000240      MU1:  NOP
3954 020700 010412      MOV    R4,(R2)                ;SET THE MAINT REG.
3955 020702 005012      CLR    (R2)                   ;THIS FETCH SHOULD CAUSE
3956                                     ;A PARITY ERROR IN GROUP
3957                                     ;ADDRESS 0 MEMORY
3958
3959 020704          MU2:
3960 020704 010437 001230      MOV    R4,$TMP2              ;REPORT ERROR. MAINTENANCE
3961                                     ;FUNCTION FAILED TO
3962                                     ;CAUSE ERROR.
3962 020710 104127      1$:  ERROR 127
3963 020712 012737 177777 032344      MOV    #-1,MANFL2
3964 020720 000500      BR     MUDONE
3965
3966 020722 022737 002420 177744      MUERR0: CMP    #2420,@#MEMERR      ;DID THE ERROR REGISTER
3967 020730 001042      BNE    69$                    ;SET PROPERLY?
3968
3969 020732 022626      64$:  CMP    (SP)+,(SP)+          ;RESET THE STACK
3970 020734 005037 177572      65$:  CLR    @#MMR0
3971 020740 005037 172516      CLR    @#MMR3
3972 020744 012737 177777 177744      MOV    #-1,@#MEMERR          ;TRY TO CLEAR THE ERROR
3973 020752 005737 177744      TST    @#MEMERR              ;REGISTER.
3974 020756 001416      BEQ    68$
3975
3976 020760          66$:
3977 020760 013737 177740 001230      MOV    @#LOADRS,$TMP2        ;ERROR REGISTER WON'T
3978 020766 013737 177742 001232      MOV    @#HIADRS,$TMP3        ;CLEAR
3979 020774 013737 177744 001234      MOV    @#MEMERR,$TMP4
3980
3981 021002 104130      67$:  ERROR 130
3982 021004 012737 177777 032324      MOV    #-1,MMRFLG            ;SIGNAL BAD REGISTER
3983 021012 000443      BR     MUDONE
3984
3985 021014 022737 177740 177740      68$:  CMP    #177740,@#LOADRS      ;SEE IF ADDRESS REGISTER
3986 021022 001356      BNE    66$                    ;UNLOCKED.
3987 021024 022737 000003 177742      CMP    #3,@#HIADRS
3988 021032 001352      BNE    66$
3989 021034 000432      BR     MUDONE
3990
3991 021036          69$:
3992 021036 012637 001230      MOV    (SP)+,$TMP2           ;REPORT ERROR REGISTER
3993 021042 005726      TST    (SP)+                 ;NOT SET AS EXPECTED.
3994 021044 013737 177740 001232      MOV    @#LOADRS,$TMP3        ;RESET THE STACK.
3995 021052 013737 177742 001234      MOV    @#HIADRS,$TMP4
3996 021060 012737 000400 001236      MOV    #400,$TMP5
    
```

```

3997 021066 012737 002420 001240      MOV    #2420,$TMP6
3998 021074 013737 177744 001242      MOV    @MMEMERR,$TMP7
3999
4000 021102 104131      70$:   ERROR    131
4001 021104 012737 177777 032344      MOV    #-1,$MANFL2      ;SIGNAL BAD REGISTER
4002 021112 012737 177777 032340      MOV    #-1,$MMRFL2
4003 021120 000705      BR     65$
4004 021122 104416      MUDONE: RSFT
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017 021124 000004      TST40: SCOPE
4018 021126 012737 000040 001274      MOV    #40,$TIMES      ;;DO 40 ITERATIONS
4019
4020      MV=$TN-1
4021 021134 012737 021544 032110      MOV    #TST41,$KAD      ;SET THE SKAD REGISTER
4022
4023 021142 113737 001102 001224      MOV    $TSTNM,$TMP0      ;IN CASE THE TEST ABORTS.
4024
4025 021150 104430      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4026 021152 104432      SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4027 021154 104434      SKPBMM      ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4028 021156 104436      SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4029 021160 104422      MMSKIP
4030
4031 021162 012700 172340      MOV    #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
4032
4033 021166 012702 172300      MOV    #KIPDR0,R2      ;TO RELOCATE EVERYTHING
4034 021172 012703 000007      MOV    #7,R3           ;THROUGH THE UNIBUS
4035 021176 005004      CLR     R4             ;MAP PASSIVELY TO MEMORY,
4036 021200 012705 170200      MOV    #MAPL00,R5      ;BY PASSIVELY IS MEANT
4037
4038 021204 012722 077406      64$:   MOV    #77406,(R2)+    ;THAT ADDRESS ARE
4039 021210 010401      MOV    R4,R1           ;RELOCATED TO THEMSELVES.
4040 021212 072127 000006      ASH    #6,R1
4041 021216 010125      MOV    R1,(R5)+
4042 021220 005025      CLR    (R5)+
4043 021222 010410      MOV    R4,(R0)
4044 021224 062720 170000      ADD    #170000,(R0)+
4045 021230 062704 000200      ADD    #200,R4
4046 021234 077315      SOB    R3,64$
4047 021236 012710 177600      MOV    #177600,(R0)
4048 021242 012712 077406      MOV    #77406,(R2)
4049
4050 021246 012737 000060 172516      MOV    #60,@MMR3      ;TURN ON THE MAP AND
4051 021254 012737 000001 177572      MOV    #1,@MMR0      ;22-BIT MODE ADDRESSING
4052 021262 012737 021342 000114      MOV    #MVERRO,@CACHVEC ;SETUP FOR THE ERROR.
    
```

```

*****
*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 .
*
*****
    
```

```

4053 021270 012737 000044 177746      MOV      #S1M0,@CONTRL ;SELECT GROUP ADDRESS
4054 021276 012704 002000              MOV      #2000,R4      ;PATTERN TO BE LOADED IN THE
4055 021302 012702 177750              MOV      #MAINT,R2     ;MAINTENANCE REG.
4056 021306 000403              BR       MV1
4057
4058              021310              LOC=      ;GET THE PC TO AN EVEN WORD BOUNDARY
4059              021310              LOC=-4&LOC
4060              021314              LOC=LOC+4
4061              021314              .=LOC
4062
4063 021314 000240              NOP
4064 021316 000240      MV1:      NOP
4065 021320 010412              MOV      R4,(R2)      ;SET THE MAINT REG.
4066 021322 005012              CLR      (R2)         ;THIS FETCH SHOULD CAUSE
4067                                     ;A PARITY ERROR IN GROUP
4068                                     ;ADDRESS 1 MEMORY
4069
4070 021324              MV2:
4071 021324 010437 001230              MOV      R4,$TMP2     ;REPORT ERROR. MAINTENANCE
4072                                     ;FUNCTION FAILED TO
4073                                     ;CAUSE ERROR.
4073 021330 104127      1$:      ERROR 127
4074 021332 012737 177777 032344      MOV      #-1,MANFL2
4075 021340 000500              BR       MVDONE
4076
4077 021342 022737 002440 177744      MVERRO: CMP      #2440,@MEMERR ;DID THE ERROR REGISTER
4078 021350 001042              BNE     69$           ;SET PROPERLY?
4079
4080 021352 022626      64$:      CMP      (SP)+,(SP)+ ;RESET THE STACK
4081 021354 005037 177572      65$:      CLR      @MMR0
4082 021360 005037 172516              CLR      @MMR3
4083 021364 012737 177777 177744      MOV      #-1,@MEMERR ;TRY TO CLEAR THE ERROR
4084 021372 005737 177744              TST     @MEMERR      ;REGISTER.
4085 021376 001416              BEQ     68$
4086
4087 021400              66$:
4088 021400 013737 177740 001230      MOV      @LOADRS,$TMP2 ;ERROR REGISTER WON'T
4089 021406 013737 177742 001232      MOV      @HIADRS,$TMP3 ;CLEAR
4090 021414 013737 177744 001234      MOV      @MEMERR,$TMP4
4091
4092 021422 104130      67$:      ERROR 130
4093 021424 012737 177777 032324      MOV      #-1,MMRFLG  ;SIGNAL BAD REGISTER
4094 021432 000443              BR       MVDONE
4095
4096 021434 022737 177740 177740      68$:      CMP      #177740,@LOADRS ;SEE IF ADDRESS REGISTER
4097 021442 001356              BNE     66$           ;UNLOCKED.
4098 021444 022737 000003 177742      CMP      #3,@HIADRS
4099 021452 001352              BNE     66$
4100 021454 000432              BR       MVDONE
4101
4102 021456              69$:
4103 021456 012637 001230      MOV      (SP)+,$TMP2  ;REPORT ERROR REGISTER
4104 021462 005726              TST     (SP)+        ;NOT SET AS EXPECTED.
4105 021464 013737 177740 001232      MOV      @LOADRS,$TMP3 ;RESET THE STACK.
4106 021472 013737 177742 001234      MOV      @HIADRS,$TMP4
4107 021500 012737 002000 001236      MOV      #2000,$TMP5
4108 021506 012737 002440 001240      MOV      #2440,$TMP6
    
```

```

4109 021514 013737 177744 001242      MOV      @MEMERR,$TMP7
4110
4111 021522 104131      70$:    ERROR 131
4112 021524 012737 177777 032344      MOV      #-1,$MANFL2      ;SIGNAL BAD REGISTER
4113 021532 012737 177777 032340      MOV      #-1,$MMRFL2
4114 021540 000705      BR       65$
4115 021542 104416      MVDONE: RSET
4116
4117      ;*****
4118      ;*TEST 41      CACHE MAINTENANCE AND ERROR REGISTERS TEST 25
4119      ;*
4120      ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
4121      ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
4122      ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
4123      ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY
4124      ;*PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE
4125      ;*LOW BYTE OF THAT DATA .
4126      ;*
4127      ;*****
4128 021544 000004      TST41:  SCOPE
4129 021546 012737 000040 001274      MOV      #40,$TIMES      ;;DO 40 ITERATIONS
4130      MW $TN-1
4131
4132 021554 012737 022164 032110      MOV      #TST42,$SKAD      ;SET THE SKAD REGISTER
4133      ;IN CASE THE TEST ABURTS.
4134 021562 113737 001102 001224      MOV      $TSTNM,$TMP0
4135
4136 021570 104430      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4137 021572 104432      SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4138 021574 104434      SKPBMN      ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4139 021576 104436      SKPSHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4140 021600 104422      MMSKIP
4141
4142 021602 012700 172340      MOV      #KIPAR0,$R0      ;SET UP MEMORY MANAGEMENT
4143      ;TO RELOCATE EVERYTHING
4144 021606 012702 172300      MOV      #KIPDR0,$R2      ;THROUGH THE UNIBUS
4145 021612 012703 000007      MOV      #7,$R3      ;MAP PASSIVELY TO MEMORY,
4146 021616 005004      CLR      $R4      ;BY PASSIVELY IS MEANT
4147 021620 012705 170200      MOV      #MAPL00,$R5      ;THAT ADDRESS ARE
4148      ;RELOCATED TO THEMSELVES.
4149 021624 012722 077406      64$:    MOV      #77406,(R2)+
4150 021630 010401      MOV      $R4,$R1
4151 021632 072127 000006      ASH      #6,$R1
4152 021636 010125      MOV      $R1,(R5)+
4153 021640 005025      CLR      (R5)+
4154 021642 010410      MOV      $R4,(R0)
4155 021644 062720 170000      ADD      #170000,(R0)+
4156 021650 062704 000200      ADD      #200,$R4
4157 021654 077315      SOB      $R3,64$
4158 021656 012710 177600      MOV      #177600,(R0)
4159 021662 012712 077406      MOV      #77406,(R2)
4160
4161 021666 012737 000060 172516      MOV      #60,@MMR3      ;TURN ON THE MAP AND
4162 021674 012737 000001 177572      MOV      #1,@MMR0      ;22-BIT MODE ADDRESSING
4163 021702 012737 021762 000114      MOV      @MWERR0,@CACHVEC      ;SETUP FOR THE ERROR.
4164 021710 012737 000030 177746      MOV      #SOM1,@CONTRL      ;SELECT GROUP DATA
    
```



```

4165 021716 012704 000020      MOV    #20,R4 ;PATTERN TO BE LOADED IN THE
4166 021722 012702 177750      MOV    #MAINT,R2 ;MAINTENANCE REG.
4167 021726 000403      BR     MW1
4168
4169          021730      LOC-   ;GET THE PC TO AN EVEN WORD BOUNDARY...
4170          021730      LOC=-4&LOC
4171          021734      LOC=LOC+4
4172          021734      -=LOC
4173
4174 021734 000240      NOP
4175 021736 000240      MW1:  NOP
4176 021740 010412      MOV    R4,(R2) ;SET THE MAINT REG.
4177 021742 005012      CLR    (R2) ;THIS FETCH SHOULD CAUSE
;A PARITY ERROR IN GROUP
;DATA 0 MEMORY
4178
4179
4180
4181 021744          MW2:
4182 021744 010437 001230      MOV    R4,$TMP2 ;REPORT ERROR. MAINTENANCE
;FUNCTION FAILED TO
;CAUSE ERROR.
4183
4184 021750 104127      1$:  ERROR 127
4185 021752 012737 177777 052344      MOV    #-1,MANFL2
4186 021760 000500      BR     MWDONE
4187
4188 021762 022737 002500 177744      MWERR0: CMP    #2500,@MEMERR ;DID THE ERROR REGISTER
4189 021770 001042      BNE   69$ ;SET PROPERLY?
4190
4191 021772 022626      64$:  CMP    (SP)+,(SP)+ ;RESET THE STACK
4192 021774 005037 177572      65$:  CLR    @MMR0
4193 022000 005037 172516      CLR    @MMR3
4194 022004 012737 177777 177744      MOV    #-1,@MEMERR ;TRY TO CLEAR THE ERROR
4195 022012 005737 177744      TST   @MEMERR ;REGISTER.
4196 022016 001416      BEQ   68$
4197
4198 022020          66$:
4199 022020 013737 177740 001230      MOV    @LOADRS,$TMP2 ;ERROR REGISTER WON'T
4200 022026 013737 177742 001232      MOV    @HIADRS,$TMP3 ;CLEAR
4201 022034 013737 177744 001234      MOV    @MEMERR,$TMP4
4202
4203 022042 104130      67$:  ERROR 130
4204 022044 012737 177777 032324      MOV    #-1,MMRFLG ;SIGNAL BAD REGISTER
4205 022052 000443      BR     MWDONE
4206
4207 022054 022737 177740 177740      68$:  CMP    #177740,@LOADRS ;SEE IF ADDRESS REGISTER
4208 022062 001356      BNE   66$ ;UNLOCKED.
4209 022064 022737 000003 177742      CMP    #3,@HIADRS
4210 022072 001352      BNE   66$
4211 022074 000432      BR     MWDONE
4212
4213 022076          69$:
4214 022076 012637 001230      MOV    (SP)+,$TMP2 ;REPORT ERROR REGISTER
4215 022102 005726      TST   (SP)+ ;NOT SET AS EXPECTED.
4216 022104 013737 177740 001232      MOV    @LOADRS,$TMP3 ;RESET THE STACK.
4217 022112 013737 177742 001234      MOV    @HIADRS,$TMP4
4218 022120 012737 000020 001236      MOV    #20,$TMP5
4219 022126 012737 002500 001240      MOV    #2500,$TMP6
4220 022134 013737 177744 001242      MOV    @MEMERR,$TMP7
    
```

CI  
CI

```

4221
4222 022142 104131
4223 022144 012737 177777 032344
4224 022152 012737 177777 032340
4225 022160 000705
4226 022162 104416
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239 022164 000004
4240 022166 012737 000040 001274
4241
4242
4243 022174 012737 022604 032110
4244
4245 022202 113737 001102 001224
4246
4247 022210 104430
4248 022212 104432
4249 022214 104434
4250 022216 104436
4251 022220 104422
4252
4253 022222 012700 172340
4254
4255 022226 012702 172300
4256 022232 012703 000007
4257 022236 005004
4258 022240 012705 170200
4259
4260 022244 012722 077406
4261 022250 010401
4262 022252 072127 000006
4263 022256 010125
4264 022260 005025
4265 022262 010410
4266 022264 062720 170000
4267 022270 062704 000200
4268 022274 077315
4269 022276 012710 177600
4270 022302 012712 077406
4271
4272 022306 012737 000060 172516
4273 022314 012737 000001 177572
4274 022322 012737 022402 000114
4275 022330 012737 000044 177746
4276 022336 012704 000100

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

:*****
:*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 .
:*
:*****
TST42: SCOPE
MOV #40,$TIMES ;;DO 40 ITERATIONS
MX=$TN-1
MOV #TST43,SKAD ;SET THE SKAD REGISTER
;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.
MMSKIP
MOV #KIPAR0,R0 ;SET UP MEMORY MANAGEMENT
;TO RELOCATE EVERYTHING
MOV #KIPDR0,R2 ;THROUGH THE UNIBUS
MOV #7,R3 ;MAP PASSIVELY TO MEMORY,
CLR R4 ;BY PASSIVELY IS MEANT
MOV #MAPL00,R5 ;THAT ADDRESS ARE
;RELOCATED TO THEMSELVES.
64$: MOV #77406,(R2)+
MOV R4,R1
ASH #6,R1
MOV R1,(R5)+
CLR (R5)+
MOV R4,(R0)
ADD #170000,(R0)+
ADD #200,R4
SOB R3,64$
MOV #177600,(R0)
MOV #77406,(R2)

MOV #60,@MMR3 ;TURN ON THE MAP AND
MOV #1,@MMR0 ;22-BIT MODE ADDRESSING
MOV #MXERRO,@CACHVEC ;SETUP FOR THE ERROR.
MOV #S1M0,@CONTRL ;SELECT GROUP DATA
MOV #100,R4 ;PATTERN TO BE LOADED IN THE
    
```

```

4277 022342 012702 177750      MOV    #MAINT,R2      ;MAINTENANCE REG.
4278 022346 000403      BR     MX1
4279
4280          022350      LOC=.                ;GET THE PC TO AN EVEN WORD BOUNDARY!
4281          022350      LOC=-4&LOC
4282          022354      LOC=LOC+4
4283          022354      .=LOC
4284
4285 022354 000240      NOP
4286 022356 000240      MX1:  NOP
4287 022360 010412      MOV    R4,(R2)      ;SET THE MAINT REG.
4288 022362 005012      CLR    (R2)        ;THIS FETCH SHOULD CAUSE
4289                                     ;A PARITY ERROR IN GROUP
4290                                     ;DATA 1 MEMORY
4291
4292          022364      MX2:
4293 022364 010437 001230      MOV    R4,$TMP2     ;REPORT ERROR. MAINTENANCE
4294                                     ;FUNCTION FAILED TO
4295                                     ;CAUSE ERROR.
4295 022370 104127      1$:  ERROR 127
4296 022372 012737 177777 032344  MOV    #-1,MANFL2
4297 022400 000500      BR     MXDONE
4298
4299 022402 022737 002600 177744  MXERRO: CMP    #2600,@MEMERR ;DID THE ERROR REGISTER
4300 022410 001042      BNE    69$         ;SET PROPERLY?
4301
4302 022412 022626      64$:  CMP    (SP)+,(SP)+ ;RESET THE STACK
4303 022414 005037 177572      65$:  CLR    @MMR0
4304 022420 005037 172516      CLR    @MMR3
4305 022424 012737 177777 177744  MOV    #-1,@MEMERR ;TRY TO CLEAR THE ERROR
4306 022432 005737 177744      TST    @MEMERR     ;REGISTER.
4307 022436 001416      BEQ    68$
4308
4309 022440      66$:
4310 022440 013737 177740 001230      MOV    @LOADRS,$TMP2 ;ERROR REGISTER WON'T
4311 022446 013737 177742 001232      MOV    @HIADRS,$TMP3 ;CLEAR
4312 022454 013737 177744 001234      MOV    @MEMERR,$TMP4
4313
4314 022462 104130      67$:  ERROR 130
4315 022464 012737 177777 032324  MOV    #-1,MMRFLG  ;SIGNAL BAD REGISTER
4316 022472 000443      BR     MXDONE
4317
4318 022474 022737 177740 177740  68$:  CMP    #177740,@LOADRS ;SEE IF ADDRESS REGISTER
4319 022502 001356      BNE    66$         ;UNLOCKED.
4320 022504 022737 000003 177742  CMP    #3,@HIADRS
4321 022512 001352      BNE    66$
4322 022514 000432      BR     MXDONE
4323
4324          022516      69$:
4325 022516 012637 001230      MOV    (SP)+,$TMP2  ;REPORT ERROR REGISTER
4326 022522 005726      TST    (SP)+       ;NOT SET AS EXPECTED.
4327 022524 013737 177740 001232      MOV    @LOADRS,$TMP3 ;RESET THE STACK.
4328 022532 013737 177742 001234      MOV    @HIADRS,$TMP4
4329 022540 012737 000100 001236      MOV    #100,$TMP5
4330 022546 012737 002600 001240      MOV    #2600,$TMP6
4331 022554 013737 177744 001242      MOV    @MEMERR,$TMP7
4332
4332

```

```

4333 022562 104131 70$: ERROR 131
4334 022564 012737 177777 032344 MOV #-1,MANFL2 ;SIGNAL BAD REGISTER
4335 022572 012737 177777 032340 MOV #-1,MMRFL2
4336 022600 000705 BR 65$
4337 022602 104416 MXDONE: RSET
4338
4339
4340 :*****
4341 :*TEST 43 CACHE ERROR REGISTER UNIBUS TIME OUT TEST
4342 :*
4343 :*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO COMPREHEND A
4344 :*CPU TO UNIBUS THROUGH THE MAP TO THE CACHE REFERENCE WHICH
4345 :*TIMES OUT IN MAIN MEMORY. MANY SUCH NON-EXISTENT MEMORY LOCATIONS
4346 :*ARE CONVIENTLY GUARENTEED TO EXIST! ALL THE ADDRESSES
4347 :*FROM 1700000 THROUGH 1777776 ARE ADDRESSES
4348 :*WHICH CAN NOT EXIST. HERE ONLY ONE OF THESE ADDRESSES, 1777776,
4349 :*WILL BE USED TO CAUSE A TIME OUT ON THE UNIBUS AN THE CONSEQUENT
4350 :*ABORT TO VECTOR ERRVEC.
4351 :*
4352 :*****
4353 022604 000004 TST43: SCOPE
4354 022606 012737 000040 001274 MOV #40,$TIMES ;;DO 40 ITERATIONS
4355 000043 MQ-$TN-1
4356 022614 012737 023234 032110 MOV #TST44,$KAD ;SET THE SKAD REGISTER
4357 ;IN CASE THE TEST ABORTS.
4358 022622 113737 001102 001224 MOVB $TSTM,$TMP0
4359 022630 012737 031764 000114 MOV #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
4360
4361 022636 104430 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4362 022640 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4363 022642 104434 SKPBMM ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4364 022644 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4365 022646 104422 MMSKIP
4366
4367 022650 012700 172340 MOV #KIPAR0,R0 ;INITIALLY PUT MEMORY
4368 022654 012701 077406 MOV #77406,R1 ;MANAGEMENT IN A 'PASSIVE'
4369 022660 012702 172300 MOV #KIPDR0,R2 ;STATE, THAT IS MAP ALL
4370 022664 012703 000010 MOV #10,R3 ;VIRTUAL ADDRESSES ON TO
4371 022670 010122 64$: MOV R1,(R2)+ ;THEMSELVES AS PHYSICAL
4372 022672 077302 SOB R3,64$ ;ADDRESSES.
4373 022674 005020 CLR (R0)+
4374 022676 012720 000200 MOV #200,(R0)+
4375 022702 012720 000400 MOV #400,(R0)+
4376 022706 012720 000600 MOV #600,(R0)+
4377 022712 012720 001000 MOV #1000,(R0)+
4378 022716 012720 001200 MOV #1200,(R0)+
4379 022722 012720 001400 MOV #1400,(R0)+
4380 022726 012710 177600 MOV #177600,(R0)
4381
4382 022732 012737 000060 172516 MOV #60,@MMR3 ;TURN ON THE MAPPING BOX
4383 022740 012737 000001 177572 MOV #1,@MMR0 ;AND 22 BIT MODE ADDRESSING.
4384 022746 012737 170000 172354 MOV #170000,@KIPAR6 ;MAKE KIPAR6 RELOCATE
4385 ;TO THE UNIBUS.
4386 022754 012737 023026 000004 MOV #MQERR,@ERRVEC ;SET UP THE TIME OUT VECTOR.
4387
4388 022762 012737 177776 170200 MOV #-2,@MAPL00 ;SET THE MAP REGISTER 0
    
```

```

4389 022770 012737 000077 170202      MOV      #77,@MAPH00
4390 022776 012700 140000      MOV      #140000,R0
4391
4392
4393
4394
4395
4396
4397
4398 023002 000240      NOP
4399 023004 005710      TST      (R0)
4400
4401 023006
4402 023006 012737 177776 001230      MOV      #-2,$TMP2
4403 023014 012737 000077 001232      MOV      #77,$TMP3
4404 023022 104132      1$:      ERROR   132
4405 023024 000502      BR       MQDONE
4406
4407 023026 032737 000020 177766      MQERR:   BII     #20,@CPUERR
4408 023034 001002      BNE     MQ2
4409 023036 000137 031736      JMP     CPSPUR
4410
4411 023042 022737 000000 177744      MQ2:     CMP     #0,@MEMERR
4412 023050 001427      BEQ     MQ3
4413
4414
4415 023052 012637 001230      MOV      (SP)+,$TMP2
4416 023056 005726      TST     (SP)+
4417 023060 013737 177740 001232      MOV      @LOADRS,$TMP3
4418 023066 013737 177742 001234      MOV      @HIADRS,$TMP4
4419 023074 012737 177776 001236      MOV      #-2,$TMP5
4420 023102 012737 000077 001240      MOV      #77,$TMP6
4421 023110 013737 177744 001242      MOV      @MEMERR,$TMP7
4422 023116 104133      1$:      ERROR   133
4423 023120 012737 177777 032340      MOV      #-1,MMRFL2
4424 023126 000401      BR       MQ4
4425
4426 023130 022626      MQ3:     CMP     (SP)+,(SP)+
4427
4428 023132 005037 177572      MQ4:     CLR     @MMR0
4429 023136 005037 172516      CLR     @MMR3
4430 023142 012737 177777 177744      MOV      #-1,@MEMERR
4431 023150 005737 177744      TST     @MEMERR
4432 023154 001416      BEQ     MQ6
4433
4434 023156
4435 023156 013737 177740 001230      MQ5:     MOV      @LOADRS,$TMP2
4436 023164 013737 177742 001232      MOV      @HIADRS,$TMP3
4437 023172 013737 177744 001234      MOV      @MEMERR,$TMP4
4438 023200 104130      1$:      ERROR   130
4439 023202 012737 177777 032324      MOV      #-1,MMRFLG
4440 023210 000410      BR       MQDONE
4441
4442 023212 022737 177740 177740      MQ6:     CMP      #177740,@LOADRS
4443 023220 001356      BNE     MQ5
4444 023222 022737 000003 177742      CMP      #3,@HIADRS
    
```

;THIS IS THE VIRTUAL ADDRESS OF THE  
 ;TEST ADDRESS. IT WILL RELOCATE  
 ;THROUGH KIPAR6 TO THE UNIBUS AS  
 ;A 000000. FROM THE UNIBUS  
 ;IT WILL BE RELOCATED THROUGH  
 ;MAP REGISTER 0 TO THE CACHE WHERE  
 ;IT WILL TRY TO REFERENCE  
 ;17777776, AND HOPEFULLY TIME OUT.  
 ;FOR SCOPING WITH AN OSCILLOSCOPE  
 ;MAKE THE REFERENCE!

;NO TIME OUT OCCURRED, REPORT  
 ;THE ERROR.

;SEE IF A TIME OUT HAS CAUSED  
 ;AN ABORT TO THIS ROUTINE.  
 ;IF NOT GO TO THE SPURIOUS  
 ;UNEXPECTED, CPU ERROR HANDLER.  
 ;OTHERWISE SEE IF THE ERROR  
 ;REGISTER GOT SET CORRECTLY.

;IF IT IS NOT SET CORRECTLY REPORT ERROR.

;RESET THE STACK

;TRY TO CLEAR THE ERROR REGISTER.

;REPORT THE FAILURE OF THE ERROR  
 ;REGISTER TO CLEAR

;SEE IF THE ADDRESS REGISTER  
 ;GOT RESET.

```

4445 023230 001352          BNE      MQ5
4446
4447 023232 104416          MQDONE: RSET
4448
4449          ;*****
4450          ;*TEST 44          CACHE CONTROL REGISTER DISABLE TRAPS TEST 1
4451          ;*
4452          ;*THIS IS A TEST OF THE CONTROL REGISTER'S ABILITY TO DISABLE A TRAP
4453          ;*OCCURRING AS THE RESULT OF A MAIN MEMORY DATA PARITY ERROR IN THE
4454          ;*UNWANTED WORD OF THE REFERENCED PAIR. THE MAINTENANCE REGISTER IS
4455          ;*USED TO FORCE AN ERROR ON THE LOW BYTE OF THE ODD WORD WHEN REFERENCING
4456          ;*THE EVEN WORD OF THAT PAIR.
4457          ;*
4458          ;*****
4459 023234 000004          TST44: SCOPE
4460 023236 012737 000040 001274          MOV      #40,$TIMES          ;;DO 40 ITERATIONS
4461          000044          KV $'N-1
4462          ;SET THE SKAD REGISTER
4463 023244 012737 023410 032110          MOV      #TST45,SKAD          ;IN CASE THE TEST ABORTS.
4464
4465 023252 113737 001102 001224          MOVB     $TSTNM,$TMP0
4466
4467 023260 104430          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4468 023262 104432          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4469 023264 104434          SKPBMM          ;IF THE MAINTENANCE REGISTER IS BAD SKIP THIS TEST.
4470 023266 104436          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4471 023270 012737 000014 177746          MOV      #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
4472 023276 052737 000001 177746          BIS      #BIT0,@#CONTRL          ;DISABLE 'WARNING' TRAPS.
4473 023304 012737 023346 000114          MOV      #KVERR,@#CACHVEC          ;SET UP FOR THE ERROR ABOUT TO BE FORCED
4474 023312 012704 040000          MOV      #40000,R4          ;PATTERN FOR THE MAINTENANCE
4475 023316 012702 177750          MOV      #MAINT,R2          ;REGISTER.
4476 023322 000402          BR       KV1
4477
4478          023324          LOC=.          ;GET THE PC TO AN EVEN WORD BOUNDARY...
4479          023324          LOC=-4&LOC
4480          023330          LOC-LOC+4
4481          023330          .-LOC
4482
4483 023330 000240          KV1:  NOP
4484 023332 010412          MOV      R4,(R2)          ;SET THE MAINTENANCE REGISTER
4485 023334 000240          NOP          ;WHEN THIS NOP IS FETCHED AN ERROR
4486 023336 005701          KV2:  TST      R1          ;WILL BE RECOGNIZED BECAUSE OF THE
4487          ;CONTENTS OF THE LOCATION KV2!
4488          ;THIS PARITY ERROR WOULD
4489          ;NORMALLY RELUT IN A TRAP BUT
4490          ;BECAUSE TRAPS HAVE BEEN DISABLED
4491          ;NONE SHOULD OCCUR!.'
4492 023340 005012          CLR      (R2)
4493 023342 000240          NOP
4494 023344 000420          BR       KVDONE          ;GOOD, NO TRAP OCCURRED
4495
4496 023346          KVERR:          ;COME HERE IF A TRAP OCCURS
4497 023346 012637 001230          MOV      (SP)+,$TMP2          ;AND REPORT THE ERROR.
4498 023352 005726          TST      (SP)+
4499 023354 013737 177746 001232          MOV      @#CONTRL,$TMP3
4500 023362 013737 177740 001234          MOV      @#LOADRS,$TMP4
    
```

4501 023370 013737 177742 001236  
 4502 023376 013737 177744 001240  
 4503 023404 104134  
 4504  
 4505 023406 104416  
 4506  
 4507  
 4508  
 4509  
 4510  
 4511  
 4512  
 4513  
 4514  
 4515  
 4516  
 4517  
 4518  
 4519 023410 000004  
 4520 023412 012737 000040 001274  
 4521 000045  
 4522  
 4523 023420 012737 023610 032110  
 4524  
 4525 023426 113737 001102 001224  
 4526  
 4527 023434 104430  
 4528 023436 104432  
 4529 023440 104434  
 4530 023442 104436  
 4531 023444 012737 000030 177746  
 4532 023452 012700 023540  
 4533 023456 005710  
 4534 023460 005710  
 4535  
 4536  
 4537 023462 032737 000010 177752  
 4538 023470 001007  
 4539  
 4540 023472 010037 001230  
 4541 023476 012737 000000 001226  
 4542 023504 104001  
 4543  
 4544 023506 104420  
 4545  
 4546 023510 052737 000001 177746 KX1:  
 4547 023516 012737 023546 000114  
 4548  
 4549 023524 012704 000400  
 4550 023530 012702 177750  
 4551 023534 000240  
 4552 023536 010412  
 4553 023540 005012 KX2:  
 4554 023542 000240  
 4555 023544 000420  
 4556

```

MOV @#HIADRS,$TMP5
MOV @#MEMERR,$TMP6
1$: ERROR 134

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.
: *
: *****
TST45: SCOPE
MOV #40,$TIMES ;:DO 40 ITERATIONS
KX $TN-1
;SET THE SKAD REGISTER
;IN CASE THE TEST ABORTS.
MOV #TST46,SKAD
MOVB $TSTNM,$TMP0
SKPBER ;:IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
SKPBCN ;:IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
SKPBMM ;:IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
SKPBHM ;:IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
MOV #SOM1,@#CONTRL ;:USE GROUP ZERO
MOV #KX2,R0 ;:MAKE KX2 A HIT IN GROUP
TST (R0) ;:ZERO.
TST (R0)
;SEE IF REFERENCE ADDRESS
;IS A HIT.
BIT #10,@#HITMIS
BNE KX1
;IF NOT ERROR!
MOV R0,$TMP2
MOV #0,$TMP1
ERROR 1
SKIPT ;:ERROR FATAL. GO TO NEXT TEST.
KX1: BIS #BIT0,@#CONTRL ;:DISABLE 'WARNING' TRAPS.
MOV #KXERR,@#CACHVEC ;:SET UP FOR ERROR WHICH
;SHOULD NOT TRAP!
;PATTERN FOR MAINT REG.
MOV #400,R4
MOV #MAINT,R2
MOV R4,(R2) ;:SET THE MAINT. REG.
CLR (R2) ;:THE FETCH OF THIS
NOP ;:INSTRUCTION SHOULD CAUSE
BR KXDONE ;:A CACHE MEMORY
;:PARITY ERROR WHICH
    
```

4557  
 4558  
 4559  
 4560  
 4561 023546  
 4562 023546 012637 001230  
 4563 023552 005726  
 4564 023554 013737 177746 001232  
 4565 023562 013737 177740 001234  
 4566 023570 013737 177742 001236  
 4567 023576 013737 177744 001240  
 4568  
 4569 023604 104134  
 4570  
 4571 023606 104416  
 4572  
 4573  
 4574  
 4575  
 4576  
 4577  
 4578  
 4579  
 4580  
 4581  
 4582  
 4583

KXERR:                   : NORMALLY SHOULD TRAP  
                          : BUT HERE NO TRAP SHOULD  
                          : OCCUR FOR TRAPS HAVE BEEN DISABLED.  
                          : A TRAP HAS ERRONEOUSLY  
                          : TAKEN PLACE, REPORT  
                          : UNABLE TO DISABLE TRAPS.  
 MOV           (SP)+, \$TMP2  
 TST           (SP)+  
 MOV           @#CONTRL, \$TMP3  
 MOV           @#LOADRS, \$TMP4  
 MOV           @#HIADRS, \$TMP5  
 MOV           @#MEMERR, \$TMP6

1\$:       ERROR    134

KXDONE: RSET

\*\*\*\*\*  
 : \*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.  
 : \*  
 : \*\*\*\*\*

4584 023610 000004  
 4585 023612 012737 000040 001274  
 4586           000046  
 4587  
 4588 023620 012737 024010 032110  
 4589  
 4590 023626 113737 001102 001224  
 4591  
 4592 023634 104430  
 4593 023636 104432  
 4594 023640 104434  
 4595 023642 104436  
 4596 023644 012737 000030 177746  
 4597 023652 012700 023740  
 4598 023656 005710  
 4599 023660 005710  
 4600  
 4601  
 4602 023662 032737 000010 177752  
 4603 023670 001007  
 4604  
 4605 023672 010037 001230  
 4606 023676 012737 000000 001226  
 4607 023704 104001  
 4608  
 4609 023706 104420  
 4610  
 4611 023710 052737 000001 177746  
 4612 023716 012737 023746 000114

TST46:   SCOPE  
 MOV       #40, \$TIMES        : DO 40 ITERATIONS  
 KZ-\$TN-1  
           : SET THE SKAD REGISTER  
           : IN CASE THE TEST ABORTS.  
 MOV       #TST47, SKAD  
 MOVB      \$TSTNM, \$TMP0  
           : IF THE ERROR REGISTER IS BAD SKIP THIS TEST.  
 SKPBER                       : IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.  
 SKPBCN                       : IF THE MAINTENANCE REGISER IS BAD SKIP TEST.  
 SKPBMN                       : IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.  
 SKPBHM                       : IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.  
 MOV       #SOM1, @#CONTRL    : USE GROUP ZERO  
 MOV       #KZ2, R0           : MAKE KZ2 A HIT IN GROUP  
 TST       (R0)               : ZERO.  
 TST       (R0)  
           : SEE IF REFERENCE ADDRESS  
           : IS A HIT.  
 BIT       #10, @#HITMIS  
 BNE       KZ1  
           : IF NOT ERROR.  
 MOV       R0, \$TMP2  
 MOV       #0, \$TMP1  
 ERROR     1  
 SKIPT                        : ERROR FATAL. GO TO NEXT TEST.  
 KZ1:     BIS       #BIT0, @#CONTRL   : DISABLE 'WARNING' TRAPS.  
           MOV       #KZERR, @#CACHVEC   : SET UP FOR ERROR WHICH



```
4613                                     ;SHOULD NOT TRAP.  
4614 023724 012704 000020             MOV #20,R4           ;PATTERN FOR MAINT REG.  
4615 023730 012702 177750             MOV #MAINT,R2  
4616 023734 000240                     NOP  
4617 023736 010412                     MOV R4,(R2)         ;SET THE MAINT. REG.  
4618 023740 005012                     CLR (R2)            ;THE FETCH OF THIS  
4619 023742 000240                     NOP                ;INSTRUCTION SHOULD CAUSE  
4620 023744 000420                     BR KZDONE           ;A CACHE MEMORY  
4621                                     ;PARITY ERROR WHICH  
4622                                     ;NORMALLY SHOULD TRAP  
4623                                     ;BUT HERE NO TRAP SHOULD  
4624                                     ;OCCUR FOR TRAPS HAVE BEEN DISABLED.  
4625  
4626 023746                               KZERR:             ;A TRAP HAS ERRONEOUSLY  
4627 023746 012637 001230             MOV (SP)+,$TMP2     ;TAKEN PLACE, REPORT  
4628 023752 005726                     TST (SP)+           ;UNABLE TO DISABLE TRAPS.  
4629 023754 013737 177746 001232     MOV @CONTRL,$TMP3  
4630 023762 013737 177740 001234     MOV @LOADRS,$TMP4  
4631 023770 013737 177742 001236     MOV @HIADRS,$TMP5  
4632 023776 013737 177744 001240     MOV @MEMERR,$TMP6  
4633  
4634 024004 104134                     1$: ERROR 134  
4635  
4636 024006 104416                     KZDONE: RSET  
4637  
4638  
4639  
4640  
4641  
4642  
4643  
4644  
4645  
4646  
4647  
4648  
4649  
4650  
4651  
4652  
4653  
4654  
4655  
4656  
4657 024010 000004  
4658 024012 012737 000040 001274  
4659                                     000047  
4660  
4661 024020 012737 024374 032110     MOV #TST50,SKAD    ;SET THE SKAD REGISTER  
4662                                     ;IN CASE THE TEST ABORTS.  
4663 024026 113737 001102 001224     MOVB $TSTNM,$TMP0  
4664  
4665 024034 104430                     SKPBR              ;IF THE ERKOR REGISTER IS BAD SKIP THIS TEST.  
4666 024036 104432                     SKPBCN            ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.  
4667 024040 104434                     SKPBMN           ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.  
4668 024042 104436                     SKPBHM           ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
```

```

4669 024044 012737 000014 177745      MOV      #MOM1,@#CONTRL      ;FORCE MISSES TO BOTH GROUPS.
4670
4671
4672 024052 012737 024126 000114      MOV      #NA3,@#CACHVEC      ;SET UP FOR THE ERROR.
4673 024060 012704 010000      MOV      #10000,R4            ;PATTERN TO BE PUT IN
4674 024064 012702 177750      MOV      #MAINT,R2           ;THE MAINT. REG.
4675 024070 000401      BR       NA1
4676
4677      024072      LOC=.      ;GET THE PC TO AN EVEN WORD BOUNDARY.
4678      024070      LOC=-4&LOC
4679      024074      LOC=LOC+4
4680      024074      .=LOC
4681
4682 024074 000240      NA1:     NOP
4683 024076 010412      MOV      R4,(R2)            ;SET THE MAINT. REG.
4684 024100 005701      NA2:     TST      R1            ;THE FETCH OF THIS INSTRUCTION
4685 024102 005012      CLR      (R2)              ;SHOULD CAUSE AN ABORT.
4686 024104 000240      NOP
4687      ;IF NONE OCCURS REPORT
4688 024106 012737 010000 001230      MOV      #10000,$TMP2      ;ERROR.
4689 024114 104127      1$:     ERROR 127
4690 024116 012737 177777 032344      MOV      #-1,MANFL2
4691 024124 000522      BR       NADONE
4692
4693
4694 024126      NA3:
4695
4696 024126 012737 024202 000114      MOV      #NA6,@#CACHVEC      ;SET UP FOR THE ERROR.
4697 024134 012704 010000      MOV      #10000,R4            ;PATTERN TO BE PUT IN
4698 024140 012702 177750      MOV      #MAINT,R2           ;THE MAINT. REG.
4699 024144 000401      BR       NA4
4700
4701      024146      LOC=.      ;GET THE PC TO AN EVEN WORD BOUNDARY.
4702      024144      LOC=-4&LOC
4703      024150      LOC=LOC+4
4704      024150      .=LOC
4705
4706 024150 000240      NA4:     NOP
4707 024152 010412      MOV      R4,(R2)            ;SET THE MAINT. REG.
4708 024154 005701      NA5:     TST      R1            ;THE FETCH OF THIS INSTRUCTION
4709 024156 005012      CLR      (R2)              ;SHOULD CAUSE AN ABORT.
4710 024160 000240      NOP
4711      ;IF NONE OCCURS REPORT
4712 024162 012737 010000 001230      MOV      #10000,$TMP2      ;ERROR!
4713 024170 104127      1$:     ERROR 127
4714 024172 012737 177777 032344      MOV      #-1,MANFL2
4715 024200 000474      BR       NADONE
4716
4717
4718 024202      NA6:
4719
4720 024202 062706 000010      ADD      #10,SP              ;RESET THE STACK.
4721 024206 022737 144404 177744      CMP      #144404,@#MEMERR    ;SEE IF THE ERROR REGISTER
4722 024214 001004      BNE     NA7                  ;IS SET CORRECTLY.
4723 024216 022737 024100 177740      CMP      #NA2,@#LOADRS      ;SEE IF THE ADDRESS REGISTER
4724 024224 001422      BEQ     NA8                  ;IS SET CORRECTLY.
    
```

```
4725
4726 024226
4727 024226 012737 144404 001230 NA7: MOV #144404,$TMP2 ;NOT SET CORRECTLY.
4728 024234 013737 177744 001232 MOV @MMEMERR,$TMP3 ;REPORT FAILURE.
4729 024242 012737 024100 001234 MOV #NA2,$TMP4
4730 024250 005037 001236 CLR $TMP5
4731 024254 013737 177740 001240 MOV @LOADRS,$TMP6
4732 024262 013737 177742 001242 MOV @HIADRS,$TMP7
4733
4734 024270 104135 1$: ERROR 135
4735
4736 024272 005037 177572 NA8: CLR @MMR0 ;TURN OFF MEMORY MANAGEMENT.
4737 024276 005037 172516 CLR @MMR3
4738 024302 012737 177777 177744 MOV #-1,@MMEMERR ;SEE IF YOU CAN CLR THE
4739 024310 005737 177744 TST @MMEMERR ;ERROR REG.
4740 024314 001416 BEQ NA10
4741
4742 024316 NA9: MOV @LOADRS,$TMP2 ;WON'T CLEAR.
4743 024316 013737 177740 001230 MOV @HIADRS,$TMP3
4744 024324 013737 177742 001232 MOV @MMEMERR,$TMP4
4745 024332 013737 177744 001234
4746
4747 024340 104130 1$: ERROR 130
4748 024342 012737 177777 032324 MOV #-1,MMRFLG
4749 024350 000410 BR NADONE
4750
4751 024352 022737 177740 177740 NA10: CMP #177740,@LOADRS ;SEE IF THE ADDRESS REGISTER
4752 024360 001356 BNE NA9 ;HAS RESET
4753 024362 022737 000003 177742 CMP #3,@HIADRS
4754 024370 001352 BNE NA9
4755
4756 024372 104416 NADONE: RSET
4757
4758
4759
```

```
*****
*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.
*
```

```
4773
4774 024374 000004 TST50: SCOPE
4775 024376 012737 000040 001274 MOV #40,$TIMES ;:DO 40 ITERATIONS
4776 000050 NB $TN-1
4777
4778 024404 012737 025064 032110 MOV #TST51,SKAD ;SET THE SKAD REGISTER
4779 ;IN CASE THE TEST ABORTS.
4780 024412 113737 001102 001224 MOVB $TSTNM,$TMP0
```

```

4781
4782 024420 104430 SKPBER ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4783 024422 104432 SKPBCN ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4784 024424 104434 SKPBMM ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
4785 024426 104436 SKPBHM ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4786 024430 104422 MMSKIP
4787
4788 024432 012700 172340 MOV #KIPAR0,R0 ;SET UP MEMORY MANAGEMENT
4789 ;TO RELOCATE EVERYTHING
4790 024436 012702 172300 MOV #KIPDR0,R2 ;THROUGH THE UNIBUS
4791 024442 012703 000007 MOV #7,R3 ;MAP PASSIVELY TO MEMORY,
4792 024446 005004 CLR R4 ;BY PASSIVELY IS MEANT
4793 024450 012705 170200 MOV #MAPL00,R5 ;THAT ADDRESS ARE
4794 ;RELOCATED TO THEMSELVES.
4795 024454 012722 077406 64$: MOV #77406,(R2)+
4796 024460 010401 MOV R4,R1
4797 024462 072127 000006 ASH #6,R1
4798 024466 010125 MOV R1,(R5)+
4799 024470 005025 CLR (R5)+
4800 024472 010410 MOV R4,(R0)
4801 024474 062720 170000 ADD #170000,(R0)+
4802 024500 062704 000200 ADD #200,R4
4803 024504 077315 SOB R3,64$
4804 024506 012710 177600 MOV #177600,(R0)
4805 024512 012712 077406 MOV #77406,(R2)
4806
4807 024516 012737 000014 177746 MOV #MOM1,@#CONTRL ;FORCE MISSES TO BOTH GROUPS.
4808
4809
4810 024524 012737 024602 000114 MOV #NB3,@#CACHVEC ;SET UP FOR THE ERROR.
4811 024532 012704 010000 MOV #10000,R4 ;PATTERN TO BE PUT IN
4812 024536 012702 177750 MOV #MAINT,R2 ;THE MAINT. REG.
4813 024542 000402 BR NB1
4814
4815 024544 LOC= ;GET THE PC TO AN EVEN WORD BOUNDARY...
4816 024544 LOC--4&LOC
4817 024550 LOC=LOC+4
4818 024550 .=LOC
4819
4820 024550 000240 NB1: NOP
4821 024552 010412 MOV R4,(R2) ;SET THE MAINT. REG.
4822 024554 005701 NB2: TST R1 ;THE FETCH OF THIS INSTRUCTION
4823 024556 005012 CLR (R2) ;SHOULD CAUSE AN ABORT.
4824 024560 000240 NOP
4825
4826 024562 012737 010000 001230 MOV #10000,$TMP2 ;IF NONE OCCURS REPORT
4827 024570 104127 1$: ERROR 127 ;ERROR!
4828 024572 012737 177777 032344 MOV #-1,MANFL2
4829 024600 000530 BR NBDONE
4830
4831
4832 024602 NB3:
4833
4834 024602 012737 000060 172516 MOV #60,@#MMR3 ;TURN ON THE MAP AND
4835 024610 012737 000001 177572 MOV #1,@#MMR0 ;22-BIT MODE ADDRESSING
4836 024616 012737 024672 000114 MOV #NB6,@#CACHVEC ;SET UP FOR ERROR
    
```

```

4837 024624 012704 010000      MOV      #10000,R4      ;PATTERN TO BE PUT IN
4838 024630 012702 177750      MOV      #MAINT,R2     ;THE MAINT. REG.
4839 024634 000401      BR       NB4
4840
4841          024636      LOC=      ;GET THE PC TO AN EVEN WORD BOUNDARY...
4842          024634      LOC=-4&LOC
4843          024640      LOC=LOC+4
4844          024640      .=LOC
4845
4846 024640 000240      NB4:     NOP
4847 024642 010412      MOV      R4,(R2)      ;SET THE MAINT. REG.
4848 024644 005701      NB5:     TST      R1      ;THE FETCH OF THIS INSTRUCTION
4849 024646 005012      CLR      (R2)         ;SHOULD CASE AN ABORT
4850 024650 000240      NOP          ;AND UNIBUS PB ASSERTED.
4851          ;NO ABORT OCCURRED!
4852 024652 012737 010000 001230      MOV      #10000,$TMP2 ;REPORT FAILURE
4853 024660 104127      1$:     ERROR    127
4854 024662 012737 177777 032330      MOV      #-1,MANFLG
4855 024670 000474      BR       NBDONE
4856
4857
4858 024672          NB6:
4859
4860 024672 062706 000010      ADD      #10,SP      ;RESET THE STACK.
4861 024676 022737 137404 177744      CMP      #137404,@MEMERR ;SEE IF THE ERROR REGISTER
4862 024704 001004      BNE     NB7          ;IS SET CORRECTLY.
4863 024706 022737 024554 177740      CMP      #NB2,@LOADRS ;SEE IF THE ADDRESS REGISTER
4864 024714 001422      BEQ     NB8          ;IS SET CORRECTLY.
4865
4866 024716          NB7:
4867 024716 012737 137404 001230      MOV      #137404,$TMP2 ;NOT SET CORRECTLY!
4868 024724 013737 177744 001232      MOV      @MEMERR,$TMP3 ;REPORT FAILURE.
4869 024732 012737 024554 001234      MOV      #NB2,$TMP4
4870 024740 005037 001236      CLR      $TMP5
4871 024744 013737 177740 001240      MOV      @LOADRS,$TMP6
4872 024752 013737 177742 001242      MOV      @HIADRS,$TMP7
4873
4874 024760 104135      1$:     ERROR    135
4875
4876 024762 005037 177572      NB8:     CLR      @MMR0      ;TURN OFF MEMORY MANAGEMENT.
4877 024766 005037 172516      CLR      @MMR3
4878 024772 012737 177777 177744      MOV      #-1,@MEMERR ;SEE IF YOU CAN CLR THE
4879 025000 005737 177744      TST      @MEMERR      ;ERROR REG.
4880 025004 001416      BEQ     NB10
4881
4882 025006          NB9:
4883 025006 013737 177740 001230      MOV      @LOADRS,$TMP2 ;WON'T CLEAR.
4884 025014 013737 177742 001232      MOV      @HIADRS,$TMP3
4885 025022 013737 177744 001234      MOV      @MEMERR,$TMP4
4886
4887 025030 104130      1$:     ERROR    130
4888 025032 012737 177777 032324      MOV      #-1,MMRFLG
4889 025040 000410      BR       NBDONE
4890
4891 025042 022737 177740 177740      NB10:   CMP      #177740,@LOADRS ;SEE IF THE ADDRESS REGISTER
4892 025050 001356      BNE     NB9          ;HAS RESET
    
```

4893 025052 022737 000003 177742  
 4894 025060 001352  
 4895  
 4896 025062 104416  
 4897  
 4898  
 4899  
 4900  
 4901  
 4902  
 4903  
 4904  
 4905  
 4906  
 4907  
 4908  
 4909  
 4910  
 4911  
 4912  
 4913  
 4914 025064 000004  
 4915 025066 012737 000040 001274  
 4916 000051  
 4917  
 4918 025074 012737 025564 032110  
 4919  
 4920 025102 113737 001102 001224  
 4921  
 4922 025110 104430  
 4923 025112 104432  
 4924 025114 104434  
 4925 025116 104436  
 4926 025120 104422  
 4927  
 4928 025122 012700 172340  
 4929  
 4930 025126 012702 172300  
 4931 025132 012703 000007  
 4932 025136 005004  
 4933 025140 012705 170200  
 4934  
 4935 025144 012722 077406  
 4936 025150 010401  
 4937 025152 072127 000006  
 4938 025156 010125  
 4939 025160 005025  
 4940 025162 010410  
 4941 025164 062720 170000  
 4942 025170 062704 000200  
 4943 025174 077315  
 4944 025176 012710 177600  
 4945 025202 012712 077406  
 4946  
 4947 025206 012737 000014 177746  
 4948

CMP #3,@#HIADRS  
 BNE NB9  
 NBDONE: PSET

```

*****
*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.
*
*****
    
```

```

TST51:  SCOPE
        MOV      #40,$TIMES      ;;DO 40 ITERATIONS
NC $TN-1
        MOV      #TST52,SKAD     ;SET THE SKAD REGISTER
        ;IN CASE THE TEST ABORTS.
        MOV      $TSTNM,$TMP0
        SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
        SKPBCN     ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
        SKPBMM     ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
        SKPBHM     ;IF THE HI*/MISS REGISTER IS BAD SKIP THIS TEST.
        MMSKIP
        MOV      #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
        ;TO RELOCATE EVERYTHING
        MOV      #KIPDR0,R2     ;THROUGH THE UNIBUS
        MOV      #7,R3          ;MAP PASSIVELY TO MEMORY,
        CLR      R4             ;BY PASSIVELY IS MEANT
        MOV      #MAPL00,R5     ;THAT ADDRESS ARE
        ;RELOCATED TO THEMSELVES.
64$:   MOV      #77406,(R2)+
        MOV      R4,R1
        ASH      #6,R1
        MOV      R1,(R5)+
        CLR      (R5)+
        MOV      R4,(R0)
        ADD      #170000,(R0)+
        ADD      #200,R4
        SOB      R3,64$
        MOV      #177600,(R0)
        MOV      #77406,(R2)
        MOV      #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
    
```

```

4949
4950 025214 012737 000060 172516      MOV    #60,@MMR3      ;TURN ON THE MAP AND
4951 025222 012737 000001 177572      MOV    #1,@MMR0      ;22-BIT MODE ADDRESSING
4952 025230 012737 025306 000114      MOV    #NC3,@CACHVEC ;SET UP FOR ERROR
4953 025236 012704 010000                MOV    #10000,R4     ;PATTERN TO BE PUT IN
4954 025242 012702 177750                MOV    #MAINT,R2    ;THE MAINT. REG.
4955 025246 000402                BR     NC1
4956
4957                025250                LOC=.                ;GET THE PC TO AN EVEN WORD BOUNDARY...
4958                025250                LOC=-4&LOC
4959                025254                LOC=LOC+4
4960                025254                .=LOC
4961
4962 025254 000240      NC1:    NOP
4963 025256 010412      MOV    R4,(R2)      ;SET THE MAINT. REG.
4964 025260 005701      NC2:    TST    R1      ;THE FETCH OF THIS INSTRUCTION
4965 025262 005012      CLR    (R2)        ;SHOULD CAUSE AN ABORT
4966 025264 000240      NOP                ;AND UNIBUS PB ASSERTED.
4967                                ;NO ABORT OCCURRED.
4968 025266 012737 010000 001230      MOV    #10000,$TMP2 ;REPORT FAILURE
4969 025274 104127      1$:    ERROR 127
4970 025276 012737 177777 032330      MOV    #-1,MANFLG
4971 025304 000526      BR     NCDONE
4972
4973
4974 025306 005037 177572      NC3:    CLR    @MMR0      ;TURN OFF MEMORY MANAGEMENT.
4975 025312 005037 172516      CLR    @MMR3
4976
4977 025316 012737 025372 000114      MOV    #NC6,@CACHVEC ;SET UP FOR THE ERROR.
4978 025324 012704 010000                MOV    #10000,R4     ;PATTERN TO BE PUT IN
4979 025330 012702 177750                MOV    #MAINT,R2    ;THE MAINT. REG.
4980 025334 000401                BR     NC4
4981
4982                025336                LOC=.                ;GET THE PC TO AN EVEN WORD BOUNDARY...
4983                025334                LOC=-4&LOC
4984                025340                LOC=LOC+4
4985                025340                .-LOC
4986
4987 025340 000240      NC4:    NOP
4988 025342 010412      MOV    R4,(R2)      ;SET THE MAINT. REG.
4989 025344 005701      NC5:    TST    R1      ;THE FETCH OF THIS INSTRUCTION
4990 025346 005012      CLR    (R2)        ;SHOULD CAUSE AN ABORT.
4991 025350 000240      NOP
4992                                ;IF NONE OCCURS REPORT
4993 025352 012737 010000 001230      MOV    #10000,$TMP2 ;ERROR!
4994 025360 104127      1$:    ERROR 127
4995 025362 012737 177777 032344      MOV    #-1,MANFL2
4996 025370 000474      BR     NCDONE
4997
4998
4999 025372      NC6:
5000
5001 025372 062706 000010                ADD    #10,SP        ;RESET THE STACK.
5002 025376 022737 167404 177744      CMP    #167404,@MEMERR ;SEE IF THE ERROR REGISTER
5003 025404 001004                BNE   NC7            ;IS SET CORRECTLY.
5004 025406 022737 025260 177740      CMP    #NC2,@LOADRS ;SEE IF THE ADDRESS REGISTER
    
```

```

5005 025414 001422          BEQ      NC8          ;IS SET CORRECTLY.
5006
5007 025416          NC7:          ;NOT SET CORRECTLY!
5008 025416 012737 167404 001230      MOV      #167404,$TMP2      ;REPORT FAILURE.
5009 025424 013737 177744 001232      MOV      @MEMERR,$TMP3
5010 025432 012737 025260 001234      MOV      #NC2,$TMP4
5011 025440 005037 001236          CLR      $TMP5
5012 025444 013737 177740 001240      MOV      @LOADRS,$TMP6
5013 025452 013737 177742 001242      MOV      @HIADRS,$TMP7
5014
5015 025460 104135          1$:      ERROR      135
5016
5017 025462 005037 177572          NC8:      CLR      @MMR0          ;TURN OFF MEMORY MANAGEMENT.
5018 025466 005037 172516          CLR      @MMR3
5019 025472 012737 177777 177744      MOV      #-1,@MEMERR      ;SEE IF YOU CAN CLR THE
5020 025500 005737 177744          TST      @MEMERR          ;ERROR REG.
5021 025504 001416          BEQ      NC10
5022
5023 025506          NC9:          ;WON'T CLEAR.
5024 025506 013737 177740 001230      MOV      @LOADRS,$TMP2
5025 025514 013737 177742 001232      MOV      @HIADRS,$TMP3
5026 025522 013737 177744 001234      MOV      @MEMERR,$TMP4
5027
5028 025530 104130          1$:      ERROR      130
5029 025532 012737 177777 032324      MOV      #-1,MMRFLG
5030 025540 000410          BR       NCDONE
5031
5032 025542 022737 177740 177740      NC10:   CMP      #177740,@LOADRS ;SEE IF THE ADDRESS REGISTER
5033 025550 001356          BNE      NC9          ;HAS RESET
5034 025552 022737 000003 177742      CMP      #3,@HIADRS
5035 025560 001352          BNE      NC9
5036
5037 025562 104416          NCDONE: RSET
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055 025564 000004          ;:*****
5056 025566 012737 000040 001274      TST52:  SCOPE          ;*TEST 52      CACHE ERROR REGISTER LOCK UP TEST 4
5057 000052          ND-$TN-1          ;*
5058          MOV      #40,$TIMES      ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
5059 025574 012737 026270 032110      MOV      #TST53,SKAD      ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
5060          ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
5061          ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
5062          ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
5063          ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
5064          ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
5065          ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
5066          ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
5067          ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
5068          ;*
5069          ;:*****

```



```

5061 025602 113737 001102 001224      MOV      $STNM,$TMP0
5062
5063 025610 104430                      SKPBER      ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
5064 025612 104432                      SKPBCN      ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
5065 025614 104434                      SKPBMM      ;IF THE MAINTENANCE REGISTER IS BAD SKIP TEST.
5066 025616 104436                      SKPBHM      ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
5067 025620 104422                      MMSKIP
5068
5069 025622 012700 172340              MOV      #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
5070                                ;TO RELOCATE EVERYTHING
5071 025626 012702 172300              MOV      #KIPDR0,R2      ;THROUGH THE UNIBUS
5072 025632 012703 000007              MOV      #7,R3           ;MAP PASSIVELY TO MEMORY,
5073 025636 005004                      CLR      R4              ;BY PASSIVELY IS MEANT
5074 025640 012705 170200              MOV      #MAPL00,R5      ;THAT ADDRESS ARE
5075                                ;RELOCATED TO THEMSELVES.
5076 025644 012722 077406              64$:  MOV      #77406,(R2)+
5077 025650 010401                      MOV      R4,R1
5078 025652 072127 000006              ASH      #6,R1
5079 025656 010125                      MOV      R1,(R5)+
5080 025660 005025                      CLR      (R5)+
5081 025662 010410                      MOV      R4,(R0)
5082 025664 062720 170000              ADD      #170000,(R0)+
5083 025670 062704 000200              ADD      #200,R4
5084 025674 077315                      SOB      R3,64$
5085 025676 012710 177600              MOV      #177600,(R0)
5086 025702 012712 077406              MOV      #77406,(R2)
5087
5088 025706 012737 000014 177746      MOV      #MOM1,#CONTRL      ;FORCE MISSES TO BOTH GROUPS.
5089
5090
5091 025714 012737 000060 172516      MOV      #60,#MMR3        ;TURN ON THE MAP AND
5092 025722 012737 000001 177512      MOV      #1,#MMR0        ;22-BIT MODE ADDRESSING
5093 025730 012737 026006 000114      MOV      #ND3,#CACHVEC    ;SET UP FOR ERROR
5094 025736 012704 010000              MOV      #10000,R4        ;PATTERN TO BE PUT IN
5095 025742 012702 177750              MOV      #MAINT,R2        ;THE MAINT. REG.
5096 025746 000402                      BR      ND1
5097
5098                                025750
5099                                025750
5100                                025754
5101                                025754
5102
5103 025754 000240                      ND1:  NOP
5104 025756 010412                      MOV      R4,(R2)          ;SET THE MAINT. REG.
5105 025760 005701                      ND2:  TST      R1          ;THE FETCH OF THIS INSTRUCTION
5106 025762 005012                      CLR      (R2)            ;SHOULD CASE AN ABORT
5107 025764 000240                      NOP                       ;AND UNIBUS PB ASSERTED.
5108                                ;NO ABORT OCCURRED!
5109 025766 012737 010000 001230      MOV      #10000,$TMP2    ;REPORT FAILURE
5110 025774 104127                      1$:  ERROR  127
5111 025776 012737 177777 032330      MOV      #-1,MANFLG
5112 026004 000530                      BR      NDDONE
5113
5114
5115 026006                      ND3:
5116

```

```

CEKBC-C PDP 11/70-74MP CACHE DIAGNOSTIC PART 1 MACY11 30A(1052) C 10 16-MAY-79 09:03 PAGE 96
CEKBCL.P11 02-MAY-79 15:02 T52 CACHE ERROR REGISTER LOCK UP TEST 4 SEQ 0119

5117 026006 012737 000060 172516 MOV #60,@MMR3 ;TURN ON THE MAP AND
5118 026014 012737 000001 177572 MOV #1,@MMR0 ;22-BIT MODE ADDRESSING
5119 026022 012737 026076 000114 MOV #ND6,@CACHVEC ;SET UP FOR ERROR
5120 026030 012704 010000 MOV #10000,R4 ;PATTERN TO BE PUT IN
5121 026034 012702 177750 MOV #MAINT,R2 ;THE MAINT. REG.
5122 026040 000401 BR ND4
5123
5124 026042 LOC= ;GET THE PC TO AN EVEN WORD BOUNDARY...
5125 026040 LOC=-4&LOC
5126 026044 LOC=LOC+4
5127 026044 .-LOC
5128
5129 026044 000240 ND4: NOP
5130 026046 010412 MOV R4,(R2) ;SET THE MAINT. REG.
5131 026050 005701 ND5: TST R1 ;THE FETCH OF THIS INSTRUCTION
5132 026052 005012 CLR (R2) ;SHOULD CASE AN ABORT
5133 026054 000240 NOP ;AND UNIBUS PB ASSERTED.
5134 ;NO ABORT OCCURRED.
5135 026056 012737 010000 001230 1$: MOV #10000,$TMP2 ;REPORT FAILURE
5136 026064 104127 ERROR 127
5137 026066 012737 177777 032330 MOV #-1,MANFLG
5138 026074 000474 BR NDDONE
5139
5140
5141 026076 ND6:
5142
5143 026076 062706 000010 ADD #10,SP ;RESET THE STACK.
5144 026102 022737 033404 177744 CMP #33404,@MEMERR ;SEE IF THE ERROR REGISTER
5145 026110 001004 BNE ND7 ;IS SET CORRECTLY.
5146 026112 022737 025760 177740 CMP #ND2,@LOADRS ;SEE IF THE ADDRESS REGISTER
5147 026120 001422 BEQ ND8 ;IS SET CORRECTLY.
5148
5149 026122 ND7: ;NOT SET CORRECTLY!
5150 026122 012737 033404 001230 MOV #33404,$TMP2 ;REPORT FAILURE.
5151 026130 013737 177744 001232 MOV @MEMERR,$TMP3
5152 026136 012737 025760 001234 MOV #ND2,$TMP4
5153 026144 005037 001236 CLR $TMP5
5154 026150 013737 177740 001240 MOV @LOADRS,$TMP6
5155 026156 013737 177742 001242 MOV @HIADRS,$TMP7
5156
5157 026164 104135 1$: ERROR 135
5158
5159 026166 005037 177572 ND8: CLR @MMR0 ;TURN OFF MEMORY MANAGEMENT.
5160 026172 005037 172516 CLR @MMR3
5161 026176 012737 177777 177744 MOV #-1,@MEMERR ;SEE IF YOU CAN CLR THE
5162 026204 005737 177744 TST @MEMERR ;ERROR REG.
5163 026210 001416 BEQ ND10
5164
5165 026212 ND9: ;WON'T CLEAR.
5166 026212 013737 177740 001230 MOV @LOADRS,$TMP2
5167 026220 013737 177742 001232 MOV @HIADRS,$TMP3
5168 026226 013737 177744 001234 MOV @MEMERR,$TMP4
5169
5170 026234 104130 1$: ERROR 130
5171 026236 012737 177777 032324 MOV #-1,MMRFLG
5172 026244 000410 BR NDDONE

```

```

5173
5174 026246 022737 177740 177740 ND10:  CMP      #177740,@#LOADRS ;SEE IF THE ADDRESS REGISTER
5175 026254 001356                BNE      ND9          ;HAS RESET
5176 026256 022737 000003 177742      CMP      #3,@#HIADRS
5177 026264 001352                BNE      ND9
5178
5179 026266 104416                NDDONE: RSET
5180
5181
5182 :*****
5183 :*TEST 53      MAIN MEMORY DATA PARITY CHECKERS LOW BYTE TEST
5184 :*
5185 :*THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS
5186 :*FOR THE LOW BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD.
5187 :*THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY
5188 :*ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY
5189 :*BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE
5190 :*THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT
5191 :*A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE
5192 :*AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY
5193 :*BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS
5194 :*SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1).
5195 :*THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA
5196 :*PARITY CHECKERS WORKS IN SUCH A WAY AS TO
5197 :*EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO
5198 :*THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO
5199 :*AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS
5200 :*ALREADY ONE THEN NO ERROR OCCURS!
5201 :*
5202 :*****
5203 026270 000004                TST53:  SCOPE
5204 026272 012737 000020 001274      MOV      #20,$TIMES      ;;DO 20 ITERATIONS
5205                000054      UA=$TN
5206
5207 026300 012737 026644 032110      MOV      #TST54,SKAD     ;SET THE SKAD REGISTER
5208                ;IN CASE THE TEST ABORTS.
5209 026306 113737 001102 001224      MOV      $TSTNM,$TMP0
5210 026314 012737 031764 000114      MOV      #SPUR,@#CACHVEC
5211
5212 026322 012737 000014 177746      MOV      #MOM1,@#CONTRL ;FORCE MISSES TO BOTH GROUPS.
5213 026330 005000                CLR      R0              ;INITIALIZE
5214
5215 026332 012737 026332 001110 UA1:  MOV      #UA1,$LPERR
5216 026340 004737 032350      JSR      PC,PARCNT      ;SEE IF THE CURRENT TEST
5217 026344 032702 000001      BIT      #BIT0,R2       ;PATTERN HAS THE PARITY BIT
5218 026350 001002                BNE      UA2             ;OFF, IF NOT GO TO NEXT
5219 026352 000137 026624      JMP      UA7             ;PATTERN
5220
5221 026356 012737 026530 000114 UA2:  MOV      #UAER1,@#CACHVEC ;SET UP FOR THE ERROR, EVEN WORD.
5222 026364 012704 010000      MOV      #10000,R4      ;THIS IS A PATTERN WHICH
5223 026370 012702 177750      MOV      #MAINT,R2      ;WHEN LOADED INTO THE
5224                ;MAINTENANCE REGISTER
5225                ;WILL FORCE AN ERROR ON
5226                ;THE MAIN MEMORY EVEN
5227 026374 012701 026524      MOV      #UATMP1,R1     ;WORD LOW BYTE
5228 026400 010011      MOV      R0,(R1)
    
```

```

5229 026402 010412      MOV      R4,(R2)      ;SET THE MAINT REG
5230 026404 021101      CMP      (R1),R1     ;THE REFERENCE TO (R1),
5231                               ;UATMP1 SHOULD CAUSE
5232                               ;AN ERROR.
5233 026406 005012      CLR      (R2)
5234 026410 005012      CLR      (R2)
5235
5236 026412              UA3:
5237                               ;THE ERROR DIDN'T OCCUR!
5238 026412 010037 001230      MOV      R0,$TMP2     ;REPORT FAILURE
5239 026416 012737 026524 001232      MOV      #UATMP1,$TMP3
5240 026424 005037 001234      CLR      $TMP4
5241 026430 104140      64$:      ERROR      140
5242
5243 026432 012737 026570 000114      UA4:      MOV      #UAER2,@#CACHVEC ;SET UP FOR THE ERROR
5244 026440 012737 026432 001110      MOV      #UA4,$LPERR   ;ON THE ODD WORD.
5245 026446 012704 040000      MOV      #40000,R4     ;THIS IS A PATTERN WHICH
5246 026452 012702 177750      MOV      #MAINT,R2     ;WHEN LOADED IN THE MAINTENANCE
5247                               ;REGISTER WILL CAUSE AN ERROR
5248 026456 012701 026526      MOV      #UATMP2,R1   ;ON THE ODD WORD, LOW BYTE.
5249 026462 010011      MOV      R0,(R1)      ;SET THE MAINT REG. AND
5250 026464 000240      NOP
5251 026466 010412      MOV      R4,(R2)      ;REFERENCE (R1), UATMP2, AND
5252 026470 021101      CMP      (R1),R1     ;CAUSE THE ERROR.
5253
5254 026472 005012      CLR      (R2)
5255 026474 005012      CLR      (R2)
5256
5257 026476              UA5:
5258                               ;THE ERROR DIDN'T OCCUR!
5259 026476 010037 001230      MOV      R0,$TMP2     ;REPORT FAILURE
5260 026502 012737 026526 001232      MOV      #UATMP2,$TMP3
5261 026510 005037 001234      CLR      $TMP4
5262 026514 104141      64$:      ERROR      141
5263
5264 026516 000442      UA6:      BR      UA7
5265
5266
5267                026520      LOC=.                ;GET THE PC TO AN EVEN WORD BOUNDARY...
5268                026520      LOC=-4&LOC
5269                026524      LOC=LOC+4
5270                026524      .=LOC
5271
5272 026524 000000      UATMP1: .WORD      0
5273 026526 000000      UATMP2: .WORD      0
5274
5275 026530              UAER1:
5276 026530 022737 104404 177744      CMP      #104404,@#MEMERR ;MAKE SURE THE ERROR
5277 026536 001402      BEQ      2$          ;REGISTER IS SET PROPERLY
5278 026540 000137 031764      1$:      JMP      SPUR
5279 026544 022737 026524 177740      2$:      CMP      #UATMP1,@#LOADRS ;MAKE SURE THE ERROR
5280 026552 001372      BNE      1$          ;OCCURRED AT THE CORRECT
5281                               ;ADDRESS.
5282 026554 022626      CMP      (SP)+,(SP)+ ;RESET THE STACK
5283 026556 012737 177777 177744      MOV      #-1,@#MEMERR ;CLEAR THE ERROR REGISTERS.
5284 026564 000137 026432      JMP      UA4         ;GO TEST THE ODD WORD

```

```

5285
5286 026570 UAER2:
5287 026570 022737 104410 177744 CMP #104410,@MEMERR ;MAKE SURE THE ERROR
5288 026576 001402 BEQ 2$ ;REGISTER IS SET PROPERLY
5289 026600 000137 031764 1$: JMP SPUR
5290 026604 022737 026526 177740 2$: CMP #UATMP2,@LOADRS ;MAKE SURE THE ERROR
5291 026612 001372 BNE 1$ ;OCCURRED AT THE CORRECT
5292 ;ADDRESS.
5293 026614 022626 CMP (SP)+,(SP)+ ;RESET THE STACK
5294 026616 012737 177777 177744 MOV #-1,@MEMERR ;CLEAR THE ERROR REGISTERS.
5295
5296 026624 022700 000377 UA7: CMP #377,R0 ;INCREMENT THE TEST PATTERN
5297 026630 001404 JAB
5298 026632 062700 000001 ADD #1,R0
5299 026636 000137 026332 JMP UA1
5300
5301 026642 104416 UA8: RSET
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324 026644 000004
5325 026646 012737 000020 001274
5326 000055
5327
5328 026654 012737 027220 032110
5329
5330 026662 113737 001102 001224
5331 026670 012737 031764 000114
5332
5333 026676 012737 000014 177746
5334 026704 005000
5335
5336 026706 012737 026706 001110 UB1: MOV #UB1,$IPERR
5337 026714 004737 032350 JSR PC,PARCNT ;SEE IF THE CURRENT TEST
5338 026720 032702 000001 BIT #BIT0,R2 ;PATTERN HAS THE PARITY BIT
5339 026724 001002 BNE UB2 ;OFF, IF NOT GO TO NEXT
5340 026726 000137 027200 JMP UB7 ;PATTERN
    
```

```

*****
*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.
*
*****
    
```

```

TST54: SCOPE
MOV #20,$TIMES ;:DO 20 ITERATIONS
UB $TN
MOV #TST55,SKAD ;SET THE SKAD REGISTER
;IN CASE THE TEST ABORTS.
MOVB $TSTNM,$TMP0
MOV #SPUR,@CACHVEC
MOV #MOM1,@CONTRL ;FORCE MISSES TO BOTH GROUPS.
CLR R0 ;INITIALIZE
UB1: MOV #UB1,$IPERR
JSR PC,PARCNT ;SEE IF THE CURRENT TEST
BIT #BIT0,R2 ;PATTERN HAS THE PARITY BIT
BNE UB2 ;OFF, IF NOT GO TO NEXT
JMP UB7 ;PATTERN
    
```

```

5341
5342 026732 012737 027104 000114 UB2: MOV #UBER1,@#CACHVEC ;SET UP FOR THE ERROR, EVEN WORD.
5343 026740 012704 020000 MOV #20000,R4 ;THIS IS A PATTERN WHICH
5344 026744 012702 177750 MOV #MAINT,R2 ;WHEN LOADED INTO THE
5345 ; MAINTENANCE REGISTER
5346 ; WILL FORCE AN ERROR ON
5347 ; THE MAIN MEMORY EVEN
5348 026750 012701 027100 MOV #UBTMP1,R1 ;WORD HIGH BYTE
5349 026754 010011 MOV R0,(R1)
5350 026756 010412 MOV R4,(R2) ;SET THE MAINT REG
5351 026760 021101 CMP (R1),R1 ;THE REFERENCE TO (R1),
5352 ;UBTMP1 SHOULD CAUSE
5353 ;AN ERROR.
5354 026762 005012 CLR (R2)
5355 026764 005012 CLR (R2)
5356
5357 026766 UB3:
5358 ;THE ERROR DIDN'T OCCUR.
5359 026766 010037 001230 MOV R0,$TMP2 ;REPORT FAILURE
5360 026772 012737 027100 001232 MOV #UBTMP1,$TMP3
5361 027000 005037 001234 CLR $TMP4
5362 027004 104142 64$: ERROR 142
5363
5364 027006 012737 027144 000114 UB4: MOV #UBER2,@#CACHVEC ;SET UP FOR THE ERROR
5365 027014 012737 027006 001110 MOV #UB4,$LPERR ;ON THE ODD WORD.
5366 027022 012704 100000 MOV #100000,R4 ;THIS IS A PATTERN WHICH
5367 027026 012702 177750 MOV #MAINT,R2 ;WHEN LOADED IN THE MAINTENANCE
5368 ;REGISTER WILL CAUSE AN ERROR
5369 027032 012701 027102 MOV #UBTMP2,R1 ;ON THE ODD WORD, LOW BYTE.
5370 027036 010011 MOV R0,(R1) ;SET THE MAINT REG. AND
5371 027040 000240 NOP
5372 027042 010412 MOV R4,(R2) ;REFERENCE (R1), UBTMP2, AND
5373 027044 021101 CMP (R1),R1 ;CAUSE THE ERROR.
5374
5375 027046 005012 CLR (R2)
5376 027050 005012 CLR (R2)
5377
5378 027052 JB5:
5379 ;THE ERROR DIDN'T OCCUR.
5380 027052 010037 001230 MOV R0,$TMP2 ;REPORT FAILURE
5381 027056 012737 027102 001232 MOV #UBTMP2,$TMP3
5382 027064 005037 001234 CLR $TMP4
5383 027070 104143 64$: ERROR 143
5384
5385 027072 000442 UB6: BR UB7
5386
5387
5388 027074 LOC=. ;GET THE PC TO AN EVEN WORD BOUNDARY.
5389 027074 LOC=-4&LOC
5390 027100 LOC=LOC+4
5391 027100 .=LOC
5392
5393 027100 000000 UBTMP1:.WORD 0
5394 027102 000000 UBTMP2:.WORD 0
5395
5396 027104 UBER1:

```

```

5397 027104 022737 104404 177744      CMP      #104404,@MEMERR      :MAKE SURE THE ERROR
5398 027112 001402                    BEQ      2$                :REGISTER IS SET PROPERLY
5399 027114 000137 031764              1$:    JMP      SPUR
5400 027120 022737 027100 177740      2$:    CMP      #UBTMP1,@LOADRS  :MAKE SURE THE ERROR
5401 027126 001372                    BNE     1$                :OCCURRED AT THE CORRECT
5402                                :ADDRESS.
5403 027130 022626                    CMP      (SP)+,(SP)+      :RESET THE STACK
5404 027132 012737 177777 177744      MOV     #-1,@MEMERR      :CLEAR THE ERROR REGISTERS.
5405 027140 000137 027006                    JMP     UB4                :GO TEST THE ODD WORD
5406
5407 027144                                UBER2:
5408 027144 022737 104410 177744      CMP     #104410,@MEMERR  :MAKE SURE THE ERROR
5409 027152 001402                    BEQ     2$                :REGISTER IS SET PROPERLY
5410 027154 000137 031764              1$:    JMP     SPUR
5411 027160 022737 027102 177740      2$:    CMP     #UBTMP2,@LOADRS :MAKE SURE THE ERROR
5412 027166 001372                    BNE     1$                :OCCURRED AT THE CORRECT
5413                                :ADDRESS.
5414 027170 022626                    CMP     (SP)+,(SP)+      :RESET THE STACK
5415 027172 012737 177777 177744      MOV     #-1,@MEMERR      :CLEAR THE ERROR REGISTERS.
5416
5417 027200 022700 177400              UB7:   CMP     #177400,R0        :INCREMENT THE TEST PATTERN
5418 027204 001404                    BEQ     JB8
5419 027206 062700 000400              ADD     #400,R0
5420 027212 000137 026706                    JMP     UB1
5421
5422 027216 104416              UB8:   RSET
5423
5424
5425 027220              TST55:
5426
5427              ;:*****
5428
5429              .SBTTL  END OF PASS ROUTINE
5430
5431              ;*INCREMENT THE PASS NUMBER ($PASS)
5432              ;*INDICATE END-OF-PROGRAM AFTER * PASSES THRU THE PROGRAM
5433              ;*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
5434              ;*IF THERES A MONITOR GO TO IT
5435              ;*IF THERE ISN'T JUMP TO LOOP
5436
5437 027220              $EOP:
5438 027220 000004              SCOPE
5439 027222 005037 001102          CLR     $STNM             ;;ZERO THE TEST NUMBER
5440 027226 005037 001274          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
5441 027232 005237 001100          INC     $PASS           ;;INCREMENT THE PASS NUMBER
5442 027236 042737 100000 001100  BIC     #100000,$PASS    ;;DON'T ALLOW A NEG. NUMBER
5443 027244 005327              DEC     (PC)+           ;;LOOP?
5444 027246 000001              $EOPCT: .WORD 1
5445 027250 003031              BGT     $DOAGN          ;;YES
5446 027252 012737              MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
5447 027254 000001              $ENDCT: .WORD 1
5448 027256 027246              $EOPCT
5449 027260 104400 027340          TYPE   $ENDMG          ;;TYPE 'END PASS #'
5450 027264 013746 001100          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
5451 027270 104410              TYPDS  ;;GO TYPE--DECIMAL ASCII WITH SIGN
5452 027272 *04400 027355          TYPE   $FNUL           ;;TYPE A NULL CHARACTER
    
```

```

5453 027276 013700 000042 $GET42: MOV @#42,R0 ;;GET MONITOR ADDRESS
5454 027302 001414 BEQ $DOAGN ;;BRANCH IF NO MONITOR
5455 027304 012703 125252 MOV #125252,R3
5456 027310 004737 032424 JSR PC,CHAINQ
5457 027314 013700 000042 MOV @#42,R0 ;;INSURE R0 CONTAINS THE MONITORS
5458 027320 001405 BEQ $DOAGN ;;RETURN ADDRESS
5459 027322 000005 RESET ;;CLEAR THE WORLD
5460 027324 004710 $ENDAD: JSR PC,(R0) ;;GO TO MONITOR
5461 027326 000240 NOP ;;SAVE ROOM
5462 027330 000240 NOP ;;FOR
5463 027332 000240 NOP ;;ACT11
5464 027334 $DOAGN:
5465 027334 000137 004176 JMP @#LOOP ;;RETURN
5466 027340 005015 047105 020104 $ENDMG: .ASCIZ <15><12>/END PASS #/
5467 027346 040520 051523 021440
5468 027354 000
5469 027355 377 377 000 $ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
5470
5471 ;;*****
5472
5473 .SBTTL SCOPE HANDLER ROUTINE
5474
5475 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
5476 ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
5477 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
5478 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
5479 ;*SW14=1 LOOP ON TEST
5480 ;*SW11=1 INHIBIT ITERATIONS
5481 ;*SW09=1 LOOP ON ERROR
5482 ;*SW08=1 LOOP ON TEST IN SWR<6:0>
5483 ;*CALL
5484 ;* SCOPE ;;SCOPE=IOT
5485
5486 027360 $SCOPE:
5487 027360 006137 177570 ROL @#SWR ;;LOOP ON PRESENT TEST?
5488 027364 100517 BMI $OVER ;;YES IF SW14=1
5489 ;#####START OF CODE FOR THE XOR TESTER#####
5490 027366 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
5491 ;;THIS INSTRUCTION TO A 'NOP' (NOP-240)
5492 027370 013746 000004 MOV @#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
5493 027374 012737 027414 000004 MOV #5$,@#ERRVEC ;;SET FOR TIMEOUT
5494 027402 005737 177060 TST @#177060 ;;TIME OUT ON XOR?
5495 027406 012637 000004 MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
5496 027412 000471 BR $SVLAD ;;GO TO THE NEXT TEST
5497 027414 022626 5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
5498 027416 012637 000004 MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
5499 027422 000431 BR 7$ ;;LOOP ON THE PRESENT TEST
5500 027424 6$:;#####END OF CODE FOR THE XOR TESTER#####
5501 027424 032737 000400 177570 BIT #BIT08,@#SWR ;;LOOP ON SPEC. TEST?
5502 027432 001412 BEQ 2$ ;;BR IF NO
5503 027434 052737 001000 177746 BIS #BIT9,@#CONTRL ;;TURN OFF CACHE
5504 027442 013746 177570 MOV @#SWR,-(SP) ;;SET DESIRED TEST NUM. FROM SWR
5505 027446 042716 000200 BIC #$$SWRMK,(SP) ;;STRIP AWAY UNDESIRED BITS
5506 027452 122637 001102 CMPB (SP)+,$STNM ;;ON THE RIGHT TEST?
5507 027456 001462 BEQ $OVER ;;BR IF YES
5508 027460 105737 001103 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
    
```



```

5509 027464 001421          BEQ      3$          ;;BR IF NO
5510 027466 123737 001115 001103  CMPB   $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
5511 027474 101015          BHI    3$          ;;BR IF NO
5512 027476 032737 001000 177570  BIT    #BIT09,@#SWR   ;;LOOP ON ERROR?
5513 027504 001404          BEQ    4$          ;;BR IF NO
5514 027506 013737 001110 001106 7$:  MOV    $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
5515 027514 000443          BR     $OVER
5516 027516 105037 001103          4$:  CLRB  $ERFLG        ;;ZERO THE ERROR FLAG
5517 027522 005037 001274          CLR   $TIMES        ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
5518 027526 000415          BR     1$          ;;ESCAPE TO THE NEXT TEST
5519 027530 032737 004000 177570 3$:  BIT    #BIT11,@#SWR   ;;INHIBIT ITERATIONS?
5520 027536 001011          BNE   1$          ;;BR IF YES
5521 027540 005737 001100          TST   $PASS         ;;IF FIRST PASS OF PROGRAM
5522 027544 001406          BEQ   1$          ;;      INHIBIT ITERATIONS
5523 027546 005237 001104          INC   $ICNT         ;;INCREMENT ITERATION COUNT
5524 027552 023737 001274 001104  CMP    $TIMES,$ICNT  ;;CHECK THE NUMBER OF ITERATIONS MADE
5525 027560 002021          BGE   $OVER        ;;BR IF MORE ITERATION REQUIRED
5526 027562 012737 000001 001104 1$:  MOV    #1,$ICNT     ;;REINITIALIZE THE ITERATION COUNTER
5527 027570 013737 027640 001274  MOV    $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
5528 027576 105237 001102          $SVLAD INCB  $STNM     ;;COUNT TEST NUMBERS
5529 027602 011637 001106          MOV   (SP),$LPADR   ;;SAVE SCOPE LOOP ADDRESS
5530 027606 011637 001110          MOV   (SP),$LPERR   ;;SAVE ERROR LOOP ADDRESS
5531 027612 005037 001276          CLR   $ESCAPE      ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
5532 027616 112737 000001 001115  MOVB  #1,$ERMAX     ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
5533 027624 013737 001102 177570 $OVER: MOV  $STNM,@#DISPLAY ;;DISPLAY TEST NUMBER
5534 027632 013716 001106          MOV   $IPADR,(SP)  ;;FUDGE RETURN ADDRESS
5535 027636 000002          RTI
5536 027640 000001          $MXCNT: 1          ;;FIXES PS
                    ;;MAX. NUMBER OF ITERATIONS

```

\*\*\*\*\*

```

5538
5539
5540 .SBTTL ERROR HANDLER ROUTINE
5541
5542 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
5543 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
5544 ;*AND GO TO ERTYPE ON ERROR
5545 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
5546 ;*SW15=1 HALT ON ERROR
5547 ;*      HALT CAN OCCUR BEFORE AND AFTER THE ERROR TYPEOUT
5548 ;*SW13=1 INHIBIT ERROR TYPEOUTS
5549 ;*SW10=1 BELL ON ERROR
5550 ;*SW09=1 LOOP ON ERROR
5551 ;*CALL
5552 ;*      ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
5553

```

```

5554 027642          $ERROR:
5555 027642 105237 001103 7$:  INCB  $ERFLG        ;;SET THE ERROR FLAG
5556 027646 001775          BEQ   7$          ;;DON'T LET THE FLAG GO TO ZERO
5557 027650 013737 001102 177570  MOV    $STNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
5558 027656 005737 177570          TST   @#SWR        ;;HALT ON ERROR = 1:
5559 027662 100001          BPL   8$          ;;BRANCH IF NO
5560 027664 000000          HALT
5561 027666 032737 002000 177570 8$:  BIT    #BIT10,@#SWR   ;;YES--HALT
5562 027674 001402          BEQ   1$          ;;BELL ON ERROR?
5563 027676 104400 001300          TYPE  $BELL        ;;NO - SKIP
5564 027702 005237 001112 1$:  INC   $ERTTL       ;;RING BELL
                    ;;COUNT THE NUMBER OF ERRORS

```

```

5565 027706 011637 001116      MOV      (SP), $ERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
5566 027712 162737 000002 001116      S      #2, $ERRPC
5567 027720 117737 151172 001114      MC/B    @ $ERRPC, $ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
5568 027726 032737 020000 177570      BIT     #BIT13, @ $SWR    ;;SKIP TYPEOUT IF SET
5569 027734 001004          BNE     2$                ;;SKIP TYPEOUTS
5570 027736 004737 132626      JSR     PC, ERTYPE        ;;GO TO USER ERROR ROUTINE
5571 027742 104400 01305          TYPE    $, $CRLF
5572 027746 005737 177570      2$:     TST     @ $SWR      ;;HALT ON ERROR
5573 027752 100001          BPL     9$                ;;SKIP IF CONTINUE
5574 027754 000000          HALT                                ;;HALT ON ERROR!
5575 027756 022737 027324 000042 9$:     CMP     # $ENDAD, 42    ;;ACT-11?
5576 027764 001001          BNE     3$                ;;BRANCH IF NO
5577 027766 000000          HALT                                ;;YES
5578 027770 032737 001000 177570 3$:     BIT     #BIT09, @ $SWR    ;;LOOP ON ERROR SWITCH SET?
5579 027776 001402          BEQ     4$                ;;BR IF NO
5580 030000 013716 001110          MOV     $LPERR, (SP)      ;;FUDGE RETURN FOR LOOPING
5581 030004 005737 001276      4$:     TST     $ESCAPE     ;;CHECK FOR AN ESCAPE ADDRESS
5582 030010 001402          BEQ     5$                ;;BR IF NONE
5583 030012 013716 001276          MOV     $ESCAPE, (SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
5584 030016          5$:
5585 030016 012737 177777 177744      MOV     #-1, @ $MEMFRR
5586 030024 005057 177766          CLR     @ $CPJERR
5587 030030 000002          RTI

```

;;\*\*\*\*\*

```

5588
5589
5590
5591      .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
5592
5593      ;*SAVE R0-R5
5594      ;*CALI :
5595      ;*      SAVREG
5596      ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
5597      ;*
5598      ;*TOP---(+16)
5599      ;* +2---(+18)
5600      ;* +4---R5
5601      ;* +6---R4
5602      ;* +8---R3
5603      ;*+10---R2
5604      ;*+12---R1
5605      ;*+14---R0
5606

```

```

5607 030032      $SAVREG:
5608 030032 010046      MOV     R0, -(SP)        ;;PUSH R0 ON STACK
5609 030034 010146      MOV     R1, -(SP)        ;;PUSH R1 ON STACK
5610 030036 010246      MOV     R2, -(SP)        ;;PUSH R2 ON STACK
5611 030040 010346      MOV     R3, -(SP)        ;;PUSH R3 ON STACK
5612 030042 010446      MOV     R4, -(SP)        ;;PUSH R4 ON STACK
5613 030044 010546      MOV     R5, -(SP)        ;;PUSH R5 ON STACK
5614 030046 016646 000022      MOV     22(SP), -(SP)    ;;SAVE PS OF MAIN FLOW
5615 030052 016646 000022      MOV     22(SP), -(SP)    ;;SAVE PC OF MAIN FLOW
5616 030056 016646 000022      MOV     22(SP), -(SP)    ;;SAVE PS OF CALL
5617 030062 016646 000022      MOV     22(SP), -(SP)    ;;SAVE PC OF CALL
5618 030066 000002          RTI

```

```

5619
5620      ;*RESTORE R0-R5

```

```

5621 ;*CALL:
5622 ;* RESREG
5623 $PRESREG:
5624 030070 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PC OF CALL
5625 030074 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PS OF CALL
5626 030100 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PC OF MAIN FLOW
5627 030104 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PS OF MAIN FLOW
5628 030110 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
5629 030112 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
5630 030114 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
5631 030116 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
5632 030120 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
5633 030122 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
5634 030124 000002 RTI
5635
5636 ;:*****
5637
5638 .SBTTL TYPE ROUTINE
5639
5640 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
5641 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
5642 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
5643 ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
5644 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
5645 ;*
5646 ;*CALL:
5647 ;*1) USING A TRAP INSTRUCTION
5648 ;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
5649 ;*OR
5650 ;* TYPE
5651 ;* MESADR
5652 ;*
5653 ;*2) USING A JSR INSTRUCTION
5654 ;* MOV PS,-(SP) ;;PUSH PROCESSOR STATUS WORD ON THE STACK
5655 ;* JSR PC,$TYPE ;;CALL TYPE ROUTINE
5656 ;* MESADDR ;;FIRST ADDRESS OF MESSAGE
5657
5658 030126 105737 001151 $TYPE: TSTR $TPFLG ;;IS THERE A TERMINAL?
5659 030132 100002 BPL 1$ ;;BR IF YES
5660 030134 000000 HALT ;;HALT HERE IF NO TERMINAL
5661 030136 000407 BR 3$ ;;LEAVE
5662 030140 010046 1$: MOV R0,-(SP) ;;SAVE R0
5663 030142 017600 000002 MOV @2(SP),R0 ;;GET ADDRESS OF ASCIZ STRING
5664 030146 112046 2$: MOV (R0)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
5665 030150 001005 BNE 4$ ;;BR IF IT ISN'T THE TERMINATOR
5666 030152 005726 TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK
5667 030154 012600 MOV (SP)+,R0 ;;RESTORE R0
5668 030156 062716 000002 3$: ADD #2,(SP) ;;ADJUST RETURN PC
5669 030162 000002 RTI ;;RETURN
5670 030164 122716 000011 4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
5671 030170 001426 BEQ 8$
5672 030172 122716 000200 CMPB #CRLF,(SP) ;;BRANCH IF NOT
5673 030176 001004 BNE 5$
5674 030200 005726 TST (SP)+ ;;POP <CR><LF> EQUIV
5675 030202 104400 001305 TYPE , $CRLF
5676 030206 000757 BR 2$ ;;GET NEXT CHARACTER
    
```

```

5677 030210 004737 030272 5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
5678 030214 123726 001150 6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
5679 030220 001352 BNE 2$ ;;IF NO GO GET NEXT CHAR.
5680 030222 013746 001146 MOV $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
5681 ;;AND THE NULL CHAR.
5682 030226 105366 000001 7$: DECB 1(SP) ;;DOES A NULL NEED TO BE TYPED?
5683 030232 002770 BLT 6$ ;;BR IF NO--GO POP THE NULL OFF OF STACK
5684 030234 004737 030272 JSR PC,$TYPEC ;;GO TYPE A NULL
5685 030240 105337 030336 DECB $CHARCNT ;;DON'T COUNT THE NULL AS A CHARACTER
5686 030244 000770 BR 7$ ;;LOOP

```

5687  
 5688 ;;HORIZONTAL TAB PROCESSOR  
 5689

```

5690 030246 112716 000040 8$: MOVB #' ,(SP) ;;REPLACE TAB WITH SPACE
5691 030252 004737 030272 9$: JSR PC,$TYPEC ;;TYPE A SPACE
5692 030256 132737 000007 030336 B1/B #7,$CHARCNT ;;BRANCH IF NOT AT
5693 030264 001372 BNE 9$ ;;TAB STOP
5694 030266 005726 TST (SP)+ ;;POP SPACE OFF STACK
5695 030270 000726 BR 2$ ;;GET NEXT CHARACTER
5696 030272 105777 150644 $TYPEC: TSTB @STPS ;;WAIT UNTIL PRINTER IS READY
5697 030276 100375 BPL $TYPEC
5698 030300 116677 000002 150636 MOVB 2(SP),@STPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
5699 030306 122766 000015 000002 CMPB #CR,2(SP) ;;BRANCH IF
5700 030314 001003 BNE 1$ ;;NOT <CR>
5701 030316 105037 030336 CLRB $CHARCNT
5702 030322 000406 BR $TYPEX
5703 030324 122766 000012 000002 1$: CMPB #LF,2(SP) ;;EXIT
5704 030332 001402 BEQ $TYPEX ;;BRANCH IF
5705 030334 105227 INCB (PC)+ ;;<LF>
5706 030336 000000 $CHARCNT: .WORD 0 ;;INC SPACE
5707 030340 000207 $TYPEX: RTS PC ;;COUNT

```

5708  
 5709  
 5710 ;;\*\*\*\*\*  
 5711

5712 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE  
 5713

5714 ;\*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT  
 5715 ;\*OCTAL (ASCII) NUMBER AND TYPE IT.  
 5716 ;\*\$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE  
 5717 ;\*CALL:

```

5718 ;* MOV NUM,-(SP) ;;NUMBER TO BE TYPED
5719 ;* TYPOS ;;CALL FOR TYPEOUT
5720 ;* .BYTE N ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
5721 ;* .BYTE M ;;M=1 OR 0
5722 ;* ;;1=TYPE LEADING ZEROS
5723 ;* ;;0=SUPPRESS LEADING ZEROS
5724 ;*

```

5725 ;\*\$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST  
 5726 ;\*\$TYPOS OR \$TYPOC

```

5727 ;*CALL:
5728 ;* MOV NUM,-(SP) ;;NUMBER TO BE TYPED
5729 ;* TYPON ;;CALL FOR TYPEOUT
5730 ;*

```

5731 ;\*\$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER  
 5732 ;\*CALL:

```

5733          : *      MOV      NUM,-(SP)      ::NUMBER TO BE TYPED
5734          : *      TYPOC                      ::CALL FOR TYPEOUT
5735
5736 030342    017646    000000          $TYPOS: MOV      @ (SP),-(SP)      ::PICKUP THE MODE
5737 030346    116637    000001    030565  MOVB     1(SP), $OFILL      ::LOAD ZERO FILL SWITCH
5738 030354    112637    030567          MOVB     (SP)+, $OMODE+1    ::NUMBER OF DIGITS TO TYPE
5739 030360    062716    000002          ADD      #2,(SP)          ::ADJUST RETURN ADDRESS
5740 030364    000406          BR      $TYPON
5741 030366    112737    000001    030565  $TYPOC: MOVB     #1, $OFILL      ::SET THE ZERO FILL SWITCH
5742 030374    112737    000006    030567  MOVB     #6, $OMODE+1    ::SET FOR SIX(6) DIGITS
5743 030402    112737    000005    030564  $TYPON: MOVB     #5, $OCNT      ::SET THE ITERATION COUNT
5744 030410    010346          MOV      R3,-(SP)        ::SAVE R3
5745 030412    010446          MOV      R4,-(SP)        ::SAVE R4
5746 030414    010546          MOV      R5,-(SP)        ::SAVE R5
5747 030416    113704    030567          MOVB     $OMODE+1,R4      ::GET THE NUMBER OF DIGITS TO TYPE
5748 030422    005404          NEG      R4
5749 030424    062704    000006          ADD      #6,R4           ::SUBTRACT IT FOR MAX. ALLOWED
5750 030430    110437    030566          MOVB     R4, $OMODE      ::SAVE IT FOR USE
5751 030434    113704    030565          MOVB     $OFILL,R4       ::GET THE ZERO FILL SWITCH
5752 030440    016605    000012          MOV      12(SP),R5       ::PICKUP THE INPUT NUMBER
5753 030444    005003          CLR      R3              ::CLEAR THE OUTPUT WORD
5754 030446    006105          $:      ROL      R5        ::ROTATE MSB INTO 'C'
5755 030450    000404          BR      3$              ::GO DO MSB
5756 030452    006105          2$:     ROL      R5        ::FORM THIS DIGIT
5757 030454    006105          ROL      R5
5758 030456    006105          ROL      R5
5759 030460    010503          MOV      R5,R3
5760 030462    006103          3$:     ROL      R3        ::GET LSB OF THIS DIGIT
5761 030464    105337    030566          DECB     $OMODF          ::TYPE THIS DIGIT?
5762 030470    100016          BPL     7$              ::BR IF NO
5763 030472    042703    177770          BIC     #177770,R3      ::GET RID OF JUNK
5764 030476    001002          BNE     4$              ::TEST FOR 0
5765 030500    005704          TST     R4              ::SUPPRESS THIS 0?
5766 030502    001403          BEQ     5$              ::BR IF YES
5767 030504    005204          4$:     INC      R4        ::DON'T SUPPRESS ANYMORE 0'S
5768 030506    052703    000060          BIS     #'0,R3          ::MAKE THIS DIGIT ASCII
5769 030512    052703    000040          5$:     BIS     #' ,R3      ::MAKE ASCII IF NOT ALREADY
5770 030516    110337    030562          MOVB     R3,8$          ::SAVE FOR TYPING
5771 030522    104400    030562          TYPE    ,8$            ::GO TYPE THIS DIGIT
5772 030526    105337    030564          7$:     DECB     $OCNT      ::COUNT BY 1
5773 030532    003347          BGT     2$              ::BR IF MORE TO DO
5774 030534    002402          BLT     6$              ::BR IF DONE
5775 030536    005204          INC     R4              ::INSURE LAST DIGIT ISN'T A BLANK
5776 030540    000744          BR      2$              ::GO DO THE LAST DIGIT
5777 030542    012605          6$:     MOV      (SP)+,R5     ::RESTORE R5
5778 030544    012604          MOV      (SP)+,R4     ::RESTORE R4
5779 030546    012603          MOV      (SP)+,R3     ::RESTORE R3
5780 030550    016666    000002    000004  MOV      2(SP),4(SP)    ::SET THE STACK FOR RETURNING
5781 030556    012616          MOV      (SP)+,(SP)
5782 030560    000002          RTI
5783 030562     000          8$:     .BYTE    0          ::RETURN
5784 030563     000          .BYTE    0          ::STORAGE FOR ASCII DIGIT
5785 030564     000          $OCNT:   .BYTE    0          ::TERMINATOR FOR TYPE ROUTINE
5786 030565     000          $OFILL:  .BYTE    0          ::OCTAL DIGIT COUNTER
5787 030566    000000          $OMODE:  .WORD    0          ::ZERO FILL SWITCH
5788

```

```

5789          ;:*****
5790          ;:
5791          .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
5792          ;:
5793          ;:*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
5794          ;:*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
5795          ;:*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
5796          ;:*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
5797          ;:*REPLACED WITH SPACES.
5798          ;:*CALL:
5799          ;:*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
5800          ;:*      TYPDS      ;;GO TO THE ROUTINE
5801
5802          $TYPDS:
5803          030570 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
5804          030572 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
5805          030574 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
5806          030576 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
5807          030600 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
5808          030602 012746 020200  MOV      #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
5809          030606 016605 000020  MOV      20(SP),R5    ;;GET THE INPUT NUMBER
5810          030612 100004      BPL      1$           ;;BR IF INPUT IS POS.
5811          030614 005405      NEG      R5           ;;MAKE THE BINARY NUMBER POS.
5812          030616 112766 000055 000001  MOVB     #'-,1(SP)    ;;MAKE THE ASCII NUMBER NEG.
5813          030624 005000      1$:     CLR      R0           ;;ZERO THE CONSTANTS INDEX
5814          030626 012703 031004      MOV      #$DBLK,R3    ;;SETUP THE OUTPUT POINTER
5815          030632 112723 000040      MOVB     #' ,(R3)+    ;;SET THE FIRST CHARACTER TO A BLANK
5816          030636 005002      2$:     CLR      R2           ;;CLEAR THE BCD NUMBER
5817          030640 016001 030774      MOV      $DTBL(R0),R1 ;;GET THE CONSTANT
5818          030644 160105      3$:     SUB      R1,R5        ;;FORM THIS BCD DIGIT
5819          030646 002402      BLT      4$           ;;BR IF DONE
5820          030650 005202      INC      R2           ;;INCREASE THE BCD DIGIT BY 1
5821          030652 030774      BR       3$
5822          030654 060105      4$:     ADD      R1,R5        ;;ADD BACK THE CONSTANT
5823          030656 005702      TST      R2           ;;CHECK IF BCD DIGIT=0
5824          030660 001002      BNE      5$           ;;FALL THROUGH IF 0
5825          030662 105716      TSTB     (SP)         ;;STILL DOING LEADING 0'S?
5826          030664 100407      BMI      7$           ;;BR IF YES
5827          030666 106316      5$:     ASLB     (SP)         ;;MSD?
5828          030670 103003      BCC      6$           ;;BR IF NO
5829          030672 116663 000001 177777  MOVB     1(SP),-1(R3)  ;;YES--SET THE SIGN
5830          030700 052702 000060      BIS      #'0,R2       ;;MAKE THE BCD DIGIT ASCII
5831          030704 052702 000040      7$:     BIS      #' ,R2       ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
5832          030710 110223      MOVB     R2,(R3)+     ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
5833          030712 005720      TST      (R0)+        ;;JUST INCREMENTING
5834          030714 020027 000010      CMP      R0,#10       ;;CHECK THE TABLE INDEX
5835          030720 002746      BLT      2$           ;;GO DO THE NEXT DIGIT
5836          030722 003002      BGT      8$           ;;GO TO EXIT
5837          030724 010502      MOV      R5,R2        ;;GET THE LSD
5838          030726 000764      BR       6$           ;;GO CHANGE TO ASCII
5839          030730 105726      8$:     TSTB     (SP)+     ;;WAS THE LSD THE FIRST NON-ZERO?
5840          030732 100003      BPL      9$           ;;BR IF NO
5841          030734 116663 177777 177776  MOVB     -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
5842          030742 105013      9$:     CLRB     (R3)       ;;SET THE TERMINATOR
5843          030744 012605      MOV      (SP)+,R5     ;;POP STACK INTO R5
5844          030746 012603      MOV      (SP)+,R3     ;;POP STACK INTO R3

```

```

5845 030750 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
5846 030752 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
5847 030754 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
5848 030756 104400      TYPE     $DBLK         ;;NOW TYPE THE NUMBER
5849 030762 016666      MOV      2(SP),4(SP)   ;;ADJUST THE STACK
5850 030770 012616      MOV      (SP)+,(SP)
5851 030772 000002      RTI
5852 030774 023420      $DTBL: 10000.         ;;RETURN TO USER
5853 030776 001750      1000.
5854 031000 000144      100.
5855 031002 000012      10.
5856 031004 000004      $DBLK: .BLKW 4
5857
5858      ;;*****
5859
5860      .SBTTL TRAP DECODER
5861
5862      ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
5863      ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5864      ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5865      ;*GO TO THAT ROUTINE.
5866
5867 031014 010046      $TRAP: MOV      R0,-(SP)      ;;SAVE R0
5868 031016 016600      MOV      2(SP),R0        ;;GET TRAP ADDRESS
5869 031022 005740      TST     -(R0)           ;;BACKUP BY 2
5870 031024 111000      MOV     (R0),R0         ;;GET RIGHT BYTE OF TRAP
5871 031026 016000      MOV     $TRPAD(R0),R0    ;;INDEX TO TABLE
5872 031032 000200      RTS     R0              ;;GO TO ROUTINE
5873
5874
5875      .SBTTL TRAP TABLE
5876
5877      ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
5878      ;*BY THE 'TRAP' INSTRUCTION.
5879
5880      :      ROUTINE
5881      :      -----
5882 031034      $TRPAD:
5883 031034 030126      $TYPE   ;;CALL=TYPE      TRAP+0(104400) TTY TYPEOUT ROUTINE
5884 031036 030366      $TYPOC  ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
5885 031040 030342      $TYPOS  ;;CALL=TYPOS     TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZEROS)
5886 031042 030402      $TYPON  ;;CALL=TYPON     TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST CALL)
5887 031044 030570      $TYPDS  ;;CALL=TYPDS     TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
5888 031046 030032      $SAVREG ;;CALL=SAVREG     TRAP+12(104412) SAVE R0-R5 ROUTINE
5889 031050 030070      $RESREG ;;CALL=RESREG     TRAP+14(104414) RESTORE R0-R5 ROUTINE
5890
5891 031052 032112      CLEAN   ;;CALL=RSET      TRAP+16(104416) GO RESET ALL REGISTERS.
5892 031054 032062      ABORTT  ;;CALL=SKIPT        TRAP+20(104420) THIS WILL SKIP TO THE NEXT TEST
5893 031056 032530      MMDRES  ;;CALL=MMSKIP     TRAP+22(104422) IF SWITCH # IS ON SKIP TO THE NEXT TEST
5894 031060 032552      MSIZER  ;;CALL=SIZE         TRAP+24(104424) DETERMINE THE HIGHEST ADDRESS IN MEMORY
5895 031062 032202      SKBADR  ;;CALL=SKPBAD       TRAP+26(104426) SKIP TEST IF ERROR ADDRESS REGISTER IS I
5896 031064 032226      SKBERR  ;;CALL=SKPBER       TRAP+30(104430) SKIP TEST IF ERROR REGISTER IS INOPERA
5897 031066 032244      SKBCNR  ;;CALL=SKPBCN       TRAP+32(104432) SKIP TEST IF CONTROL REGISTER IS INOPERA
5898 031070 032262      SKBMNR  ;;CALL=SKPBMN       TRAP+34(104434) SKIP TEST IF MAINTENANCE REGISTER IS INO
5899 031072 032300      SKBHMR  ;;CALL=SKPBHM       TRAP+36(104436) SKIP TEST IF HIT/MISS REGISTER IS IN OPE
5900
    
```

```

5901      ;:*****
5902
5903      .SBTTL  POWER DOWN AND UP ROUTINES
5904
5905      :POWER DOWN ROUTINE
5906 031074 012737 031222 000024 $PWRDN: MOV  $SILLUP,@#PWRVEC  ;;SET FOR FAST UP
5907 031102 012737 000340 000026      MOV  #340,@#PWRVEC+2  ;;PRIO:7
5908 031110 010046      MOV  R0,-(SP)      ;;PUSH R0 ON STACK
5909 031112 010146      MOV  R1,-(SP)      ;;PUSH R1 ON STACK
5910 031114 010246      MOV  R2,-(SP)      ;;PUSH R2 ON STACK
5911 031116 010346      MOV  R3,-(SP)      ;;PUSH R3 ON STACK
5912 031120 010446      MOV  R4,-(SP)      ;;PUSH R4 ON STACK
5913 031122 010546      MOV  R5,-(SP)      ;;PUSH R5 ON STACK
5914 031124 010637 031226      MOV  SP,$SAVR6      ;;SAVE SP
5915 031130 012737 031142 000024      MOV  #PWRUP,@#PWRVEC  ;;SET UP VECTOR
5916 031136 000000      HALT
5917 031140 000776      BR   .-2          ;;HANG UP
5918
5919      :POWER UP ROUTINE
5920 031142 013706 031226 $PWRUP: MOV  $SAVR6,SP      ;;GET SP
5921 031146 005037 031226      CLR  - $SAVR6      ;;WAIT LOOP FOR THE TTY
5922 031152 005237 031226 1$:      INC  $SAVR6      ;;WAIT FOR THE INC
5923 031156 001375      BNE  1$          ;;OF WORD
5924 031160 012605      MOV  (SP)+,R5      ;;POP STACK INTO R5
5925 031162 012604      MOV  (SP)+,R4      ;;POP STACK INTO R4
5926 031164 012603      MOV  (SP)+,R3      ;;POP STACK INTO R3
5927 031166 012602      MOV  (SP)+,R2      ;;POP STACK INTO R2
5928 031170 012601      MOV  (SP)+,R1      ;;POP STACK INTO R1
5929 031172 012600      MOV  (SP)+,R0      ;;POP STACK INTO R0
5930 031174 012737 031074 000024      MOV  #PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
5931 031202 012737 000340 000026      MOV  #340,@#PWRVEC+2  ;;PRIO:7
5932 031210 104400      TYPE          ;;REPORT THE POWER FAILURE
5933 031212 033403 $PWRMG: .WORD  POWERM      ;;POWER FAIL MESSAGE POINTER
5934 031214 012716      MOV  (PC)+,(SP)    ;;RESTART AT START
5935 031216 003014 $PWRAD: .WORD  START      ;;RESTART ADDRESS
5936 031220 000002      RTI
5937 031222 000000 $SILLUP: HALT          ;;THE POWER UP SEQUENCE WAS STARTED
5938 031224 000776      BR   .-2          ;; BEFORE THE POWER DOWN WAS COMPLETE
5939 031226 000000 $SAVR6: 0          ;;PUT THE SP HERE
5940      ;:*****
5941
5942      .SBTTL  ROUTINE TO SIZE MEMORY
5943
5944      ;*CALL:
5945      ;*      JSR   PC,$SIZE
5946      ;*      RETURN
5947      ;*$LSTAD WILL CONTAIN:
5948      ;*      WITH KT11 OPTION      -- LAST VIRTUAL ADDRESS OF THE LAST BANK
5949      ;*      WITHOUT KT11 OPTION   -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
5950      ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
5951      ;*$KT11 IS THE MEMORY MANAGEMENT KEY
5952      ;*$BIT07 = 0 DON'T USE MEMORY MANAGEMENT
5953      ;*      MUST BE SETUP BEFORE THE CALL
5954      ;*$BIT15 0 DON'T HAVE MEMORY MANAGEMENT OPTION
5955      ;*      DETERMINED BY ROUTINE
5956      ;* --NOTE--
    
```



```

5957      ;*THIS ROUTINE SUPPORTS PDP 11/74.
5958      ;*IF ACTUAL MEMORY IS LESS THAN THAT INDICATED BY THE SIZE REGISTER
5959      ;*AND A REFERENCE IS MADE TO A MEMORY ADDRESS THAT IS GREATER THAN
5960      ;*ACTUAL MEMORY BUT LESS THAN SIZE REGISTER ((INDICATED), THEN A
5961      ;*MEMORY REFERENCE TIMEOUT TO VECTOR 114 WILL OCCUR.
5962
5963 031230 010046      $$SIZE: MOV     R0,-(SP)      ;;SAVE R0 ON THE STACK
5964 031232 010146      MOV     R1,-(SP)      ;;SAVE R1 ON THE STACK
5965 031234 010246      MOV     R2,-(SP)      ;;SAVE R2 ON THE STACK
5966 031236 010346      MOV     R3,-(SP)      ;;SAVE R3 ON THE STACK
5967 031240 013746 000004  MOV     @WERRVEC,-(SP) ;;SAVE PRESENT ERROR VECTOR PS & PC
5968 031244 013746 000006  MOV     @WERRVEC+2,-(SP)
5969 031250 013746 000114  MOV     @#114,-(SP)    ;;SAVE PRESENT PARITY VECOT PS & PC
5970 031254 013746 000116  MOV     @#116,-(SP)
5971 031260 010600      MOV     SP,R0          ;;SAVE THE STACK POINTER
5972 031262 013737 177776 000006  MOV     @#PS,@WERRVEC+2 ;;SET ERRVEC PS TO PRESENT PS
5973 031270 012701 003776      MOV     #3776,R1      ;;SETUP ADDRESS
5974 031274 105727      TSTB    (PC)+         ;;USE MEMORY MANAGEMENT?
5975 031276 000200      $KT11: .WORD 200      ;;SET TO USE MEMORY MANAGEMENT
5976 031300 100065      BPL     $SCORE        ;;BR IF NO
5977 031302 012737 031446 000004  MOV     #$SKTNEX,@WERRVEC ;;SET FOR TIMEOUT
5978 031310 005737 177572      TST     @#SR0         ;;KT11 ARE YOU THERE?
5979 031314 052737 100000 031276  BIS     #100000,$KT11  ;;YES--SET KT11 KEY
5980 031322 005046      CLR     -(SP)         ;;INITIALIZE FOR 'PAR' LOADING
5981 031324 012702 172340      MOV     #KIPAR0,R2    ;;ADDRESS OF FIRST 'PAR'
5982 031330 012703 000010      MOV     #^D8,R3       ;;LOAD EIGHT 'PAR.'S' AND EIGHT 'PDR.'S'
5983 031334 012762 077406 177740 1$:  MOV     #77406,-40(R2) ;;PDR = 4K, UP, READ/WRITE
5984 031342 011622      MOV     (SP),(R2)+    ;;LOAD 'PAR'
5985 031344 062716 000200      ADD     #200,(SP)     ;;UPDATE FOR NEXT 'PAR'
5986 031350 077307      SOB     R3,1$        ;;LOOP UNTIL ALL EIGHT ARE LOADED
5987 031352 012742 177600      MOV     #177600,-(R2) ;;SETUP KIPAR7 FOR I/O
5988 031356 005042      CLR     -(R2)        ;;SETUP KIPAR6 FOR TESTING
5989 031360 012737 031376 000004  MOV     #2$,@WERRVEC  ;;CATCH TIMEOUT IF NO SR3
5990 031366 012737 000020 172516  MOV     #20,@#SR3    ;;FNABLE 22-BIT ADDRESSING
5991 031374 000401      BR      3$           ;;THIS PDP-11 HAS A SR3 REG.
5992 031376 022626      2$:  CMP     (SP)+,(SP)+ ;;CLEAN OFF THE STACK--NO SR3.
5993 031400 005237 177572      3$:  INC     @#SR0        ;;TURN ON MEMORY MANAGEMENT
5994 031404 012737 031436 000004  MOV     #$SKTOUT,@WERRVEC ;;SET FOR TIME OUT
5995 031412 012737 031560 000114  MOV     #$SMTMOUT,@#114 ;;SET FOR MEM REF TIMEOUT
5996 031420 005737 143776      4$:  TST     @#143776    ;;TRAP ON NON-EX-MEM
5997 031424 062712 000040      ADD     #40,(R2)     ;;MAKE A 1K STEP
5998 031430 023712 172356      CMP     @#KIPAR7,(R2) ;;LAST ONE?
5999 031434 101371      BHI     4$           ;;NO--TRY IT
6000 031436 011202      $KTOUT: MOV     (R2),R2  ;;GET LAST BANK+1
6001 031440 005037 177572      CLR     @#SR0        ;;TURN OFF MEMORY MANAGEMENT
6002 031444 000421      BR      $$SIZEX
6003 031446 042737 100000 031276  $KTNEX: BIC     #100000,$KT11 ;;KT11 NON-EXISTENT
6004 031454 012737 031504 000004  $SCORE: MOV     #$SCROUT,@WERRVEC ;;SET FOR TIMEOUT
6005 031462 005002      CLR     R2           ;;SET UP BANK
6006 031464 062701 004000      1$:  ADD     #4000,R1    ;;INCREMENT BY 1K
6007 031470 062702 000040      ADD     #40,R2       ;;1K STEP
6008 031474 005711      TST     (R1)         ;;TRAP ON TIME OUT
6009 031476 022701 177776      CMP     #177776,R1   ;;LAST ONE
6010 031502 001370      BNE     1$          ;;NO--TRY AGAIN
6011 031504 162701 004000      $CROUT: SUB     #4000,R1
6012 031510 162702 000040      $SIZEX: SUB     #40,R2      ;;DROP BACK
    
```

```

6013 031514 010006      MOV      R0,SP          ;;RESTORE THE STACK
6014 031516 012637 000116  MOV      (SP)+,@#116    ;;RESTOR PARITY VECTOR
6015 031522 012637 000114  MOV      (SP)+,@#114
6016 031526 012637 000006  MOV      (SP-,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
6017 031532 012637 000004  MOV      (SP)+,@#ERRVEC
6018 031536 010137 031612  MOV      R1,$LSTAD      ;;LAST ADDRESS
6019 031542 010237 031614  MOV      R2,$LSTBK      ;;LAST BANK
6020 031546 012603      MOV      (SP)+,R3       ;;RESTORE R3
6021 031550 012602      MOV      (SP)+,R2       ;;RESTORE R2
6022 031552 012601      MOV      (SP)+,R1       ;;RESTORE R1
6023 031554 012600      MOV      (SP)+,R0       ;;RESTORE R0
6024 031556 000207      RTS      PC
6025 031560 032737 000001 177744 $MIMOUT: BIT  #BIT0,@#MEMERR ;;MAKE SURE TRAP TO 114 IS DJE
6026 031566 001005      BNE     1$             ;;TO MEMORY REFERENCE TIMEOUT
6027                                     ;;IF NOT, IS IT AN ABORT?
6028 031570 032737 100000 177744  BIT  #BIT15,@#MEMERR ;;CPU ABORT?
6029 031576 001001      BNE     1$             ;;IF YES, EXIT OUT
6030 031600 000002      RTI                                     ;;IF NOT, CONTINUE
6031 031602 012737 177777 177744 1$: MOV  #-1,@#MEMERR ;;CLEAR THE MEM ERROR REG
6032 031610 000712      BR      $KTOUT
6033 031612 000000      $LSTAD: .WORD 0        ;;CONTAINS THE LAST ADDRESS
6034 031614 000000      $LSTBK: .WORD 0        ;;CONTAINS THE LAST BANK
6035
6036
6037                                     ;;*****
6038
6039 .SBTTL DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
6040
6041 ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
6042 ;*UNSIGNED OCTAL ASCII NUMBER.
6043 ;*CALL
6044 ;*   MOV      #PNTR,-(SP) ;;: POINTER TO LOW WORD OF BINARY NUMBER
6045 ;*   JSR      PC,@#$DB20 ;;: CALL THE ROUTINE
6046 ;*   RETURN   ;;: THE ADDRESS OF THE FIRST ASCII CHAR. IS ON THE STACK
6047
6048
6049 031616 104412      $DB20: SAVREG          ;;: SAVE ALL REGISTERS
6050 031620 016601 000002  MOV      2(SP),R1      ;;: PICKUP THE POINTER TO LOW WORD
6051 031624 012705 031735  MOV      #SOCTVL+13.,R5 ;;: POINTER TO DATA TABLE
6052 031630 012704 000014  MOV      #12.,R4       ;;: DO ELEVEN CHARACTERS
6053 031634 012703 177770  MOV      #^C7,R3       ;;: MASK
6054 031640 012100  MOV      (R1)+,R0      ;;: LOWER WORD
6055 031642 012101  MOV      (R1)+,R1      ;;: HIGH WORD
6056 031644 005002  CLR      R2            ;;: TERMINATOR
6057 031646 110245 1$: MOVB  R2,-(R5)      ;;: PUT CHARACTER IN DATA TABLE
6058 031650 010002  MOV      R0,R2         ;;: GET THIS DIGIT
6059 031652 005304  DEC      R4            ;;: COUNT THIS CHARACTER
6060 031654 003007  BGT     3$            ;;: BR IF NOT THE LAST DIGIT
6061 031656 001405  BEQ     2$            ;;: BR IF IT IS THE LAST DIGIT
6062 031660 005205  INC      R5            ;;: ALL DIGITS DONE-ADJUST POINTER FOR FIRST
6063 031662 010566 000002  MOV      R5,2(SP)     ;;: ASCII CHAR. & PUT IT ON THE STACK
6064 031666 104414  RESREG          ;;: RESTORE ALL REGISTERS
6065 031670 000207  RTS      PC           ;;: RETURN TO USER
6066 031672 006203 2$: ASR   R3            ;;: POSITION THE MASK FOR THE LAST DIGIT
6067 031674 006001 3$: ROR   R1            ;;: POSITION THE BINARY NUMBER FOR
6068 031676 006000  ROR    R0            ;;: THE NEXT OCTAL DIGIT
    
```

```

6069 031700 006001          ROR    R1
6070 031702 006000          ROR    R0
6071 031704 006001          ROR    R1
6072 031706 006000          ROR    R0
6073 031710 040302          BIC    R3,R2          ;;MASK OUT ALL JUNK
6074 031712 062702 000060  ADD    #'0,R2        ;;MAKE THIS CHAR. ASCII
6075 031716 000753          BR     1$            ;;GO PUT IT IN THE DATA TABLE
6076 031720 000016          $OCTVL: .BLKB 14.    ;;RESERVE DATA TABLE
6077
6078          ;THIS ROUTINE IS CALLED BY UNEXPECTED TRAPS TO VECTOR ERRVEC.
6079          ;THE ERROR IS REPORTED AND CONTROL IS TRANSFERRED BACK TO THE TEST
6080          ;FOLLOWING THE ONE THAT WAS INTERRUPTED WHEN THE ERROR OCCURRED!
6081 031736 011637 001226  CPSPUR: MOV    (SP), $TMP1
6082 031742 012737 031760 001230  MOV    #1$, $TMP2
6083 031750 013737 177766 001232  MOV    @#CPUERR, $TMP3
6084 031756 022626          CMP    (SP)+, (SP)+  ;RESET THE STACK
6085 031760 104150          1$:    ERROR    150
6086 031762 104420          SKIPT
6087
6088          ;THIS ROUTINE HANDLE UNEXPECTED TRAPS TO #CACHVEC.
6089 031764 012737 032054 000114  SPUR:  MOV    #10$, @#CACHVEC
6090 031772 013700 177744          MOV    @#MEMERR, R0
6091 031776 032700 000014          BIT    #14, R0      ;SEE IF IT WAS A MAIN MEMORY PARITY ERROR.
6092 032002 001403          BEQ    9$
6093 032004 013700 177740          MOV    @#LOADRS, R0 ;IF IT WAS THEN THE BAD PARITY IS
6094 032010 005710          TST    (R0)         ;CACHED AND MUST BE PURGED.!!!!
6095 032012 012737 031764 000114  9$:    MOV    #SPUR, @#CACHVEC
6096 032020 013737 177744 001234  MOV    @#MEMERR, $TMP4 ;TRAP HERE IF AN UNEXPECTED
6097 032026 013737 177740 001226  MOV    @#LOADRS, $TMP1 ;ERROR, PARITY, OCCURS.
6098 032034 013737 177742 001230  MOV    @#HIADRS, $TMP2
6099 032042 011637 001232          MOV    (SP), $TMP3
6100 032046 022626          CMP    (SP)+, (SP)+
6101 032050 104014          1$:    ERROR    14
6102 032052 104420          SKIPT
6103 032054 022626          10$:   CMP    (SP)+, (SP)+  ;?????
6104 032056 000137 032012          JMP    9$
6105
6106          ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL SKIPT.
6107          ;IT TELLS THE USER THAT THE CURRENT TEST HAS BEEN
6108          ;ABORTED AND THAT CONTROL IS BEING PASSED TO THE NEXT TEST.
6109 032062 011637 001226  ABORTT: MOV    (SP), $TMP1
6110 032066 112737 000015 001114  MOVB   #15, $ITEMB
6111 032074 022626          CMP    (SP)+, (SP)+
6112 032076 004737 032626          JSR    PC, ERTYPE
6113 032102 104416          RSET
6114 032104 000177 000000          JMP    @SKAD        ;GO TO @SKAD, WHICH SHOULD
6115          ;BE SET TO THE
6116 032110 000000          SKAD:  .WORD    0    ;ADDRESS OF THE NEXT TEST.
6117
6118          ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL RSET. IT CLEARS ALL
6119          ;THE IMPORTANE REGISTERS AND RESETS THE STACK.
6120          CLEAN:
6121 032112
6122
6123 032112 012737 031764 000114  MOV    #SPUR, @#CACHVEC
6124 032120 012737 031736 000004  MOV    #CPSPUR, @#ERRVEC
    
```

```

6125 032126 011637 032200      MOV      (SP),BACKAD
6126 032132 012706 001100      MOV      #STACK,SP
6127 032136 005037 177750      CLR      @MMAINI      ;CLEAR ALL CONTROL AND ERROR
6128 032142 005037 177572      CLR      @MMR0        ;REGISTERS.
6129 032146 005037 172516      CLR      @MMR3
6130 032152 005037 177746      CLR      @CONTRL
6131 032156 012737 177777 177744  MOV      #-1,@MMEMERR
6132 032164 005037 177766      CLR      @CPUERR
6133 032170 005037 177776      CLR      @PSW
6134 032174 000177 000000      JMP      @BACKAD
6135 032200 000000      BACKAD: .WORD      0
    
```

;COME HERE TO TEST THE REGISTER FLAGS AND USE THEM TO DETERMINE WHETHER  
 ;OR NOT TO SKIP A TEST WHICH RELIES ON THE FUNCTIONALLITY OF THAT REGISTER  
 ;TO BE PROPERLY RUN.

;THESE ROUTINES ARE CALLED BY THE TRAP CATCHER CALLS:  
 : SKPBAD SKIPT IF BAD ERROR ADDRESS REGISTER  
 : SKPBER SKIPT IF BAD ERROR REGISTER  
 : SKPBCN SKIPT IF BAD CONTROL REGISTER  
 : SKPMN SKIPT IF BAD MAINTENANCE REGISTER  
 : SKPBHM SKIPT IF BAD HIT/MISS REGISTER

```

6148
6149 032202 005737 032320      SKBADR: TST      LOAFLG
6150 032206 001004          BNE      1$
6151 032210 005737 032322      TST      HIAFLG
6152 032214 001001          BNE      1$
6153 032216 000002          RTI
6154 032220 104400      1$:      TYPE
6155 032222 034365          .WORD   ADRNG
6156 032224 000433          BR      SKRNG
6157
6158 032226 005737 032324      SKBERR: TST      MMRFLG
6159 032232 001001          BNE      1$
6160 032234 000002          RTI
6161 032236 104400      1$:      TYPE
6162 032240 034475          .WORD   ERRNG
6163 032242 000424          BR      SKRNG
6164
6165 032244 005737 032326      SKBCNR: TST      CONFLG
6166 032250 001001          BNE      1$
6167 032252 000002          RTI
6168 032254 104400      1$:      TYPE
6169 032256 034575          .WORD   CNRNG
6170 032260 000415          BR      SKRNG
6171
6172 032262 005737 032330      SKBMNR: TST      MANFLG
6173 032266 001001          BNE      1$
6174 032270 000002          RTI
6175 032272 104400      1$:      TYPE
6176 032274 034677          .WORD   MNRNG
6177 032276 000406          BR      SKRNG
6178
6179 032300 005737 032332      SKBMMR: TST      HIMFLG
6180 032304 001001          BNE      1$
    
```

6181 032306 000002  
6182 032310 104400  
6183 032312 035005  
6184  
6185 032314 022626  
6186 032316 104420  
6187  
6188 032320 000000  
6189 032322 000000  
6190 032324 000000  
6191 032326 000000  
6192 032330 000000  
6193 032332 000000  
6194 032334 000000  
6195 032336 000000  
6196 032340 000000  
6197 032342 000000  
6198 032344 000000  
6199 032346 000000  
6200  
6201  
6202  
6203  
6204  
6205  
6206  
6207  
6208  
6209 032350 012701 000001  
6210 032354 005002  
6211 032356 030100  
6212 032360 001401  
6213 032362 005202  
6214 032364 006301  
6215 032366 103373  
6216 032370 000207  
6217  
6218  
6219  
6220  
6221  
6222  
6223  
6224  
6225  
6226 032372 005037 177750  
6227 032376 017700 146536  
6228 032402 104416  
6229 032404 005003  
6230 032406 042700 000200  
6231 032412 022700 000003  
6232 032416 001032  
6233 032420 104400  
6234 032422 033340  
6235 032424 012704 002734  
6236 032430 012701 052710

```
RTI  
1$: TYPE  
      .WORD HMRNG  
  
SKRNG: CMP (SP)+,(SP)+      ;RESET THE STACK AND GO TO THE  
      SKIPT                  ;NEXT TEST!!!!  
  
LOAFLG: .WORD 0              ;THESE ARE FLAGS USED TO DESIGNATE  
HIAFLG: .WORD 0              ;EITHER A GOOD OR A BAD REGISTER.  
MMRFLG: .WORD 0              ;GOOD WILL BE DESIGNATED BY A  
CONFLG: .WORD 0              ;0 BAD BY A NOT ZERO..  
MANFLG: .WORD 0  
HIMFLG: .WORD 0  
LOAFL2: .WORD 0  
HIAFL2: .WORD 0  
MMRFL2: .WORD 0  
CONFL2: .WORD 0  
MANFL2: .WORD 0  
HIMFL2: .WORD 0  
  
;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.  
PARCNT: MOV #1,R1  
      CLR R2  
1$: BIT R1,R0  
      BEQ 2$  
      INC R2  
2$: ASL R1  
      BCC 1$  
      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.  
RESMON: CLR @MAINT  
      MOV @STKB,R0  
      RSET  
      CLR R3  
      BIC #BIT7,R0          ;GET THE CHARACTER, INITIALIZE THE REGISTERS  
      CMP #3,R0            ;AND SEE IF THE CHARACTER WAS ^C.  
      BNE NOCNC            ;BRANCH AND GO TO NEXT TEST IF NOT.  
      TYPE                 ;FCHOE THE CONTROL-C AS '^C'  
      .WORD CONCS  
CHAINQ: MOV #^D1500,R4      ;AND RESTORE THE MONITOR.  
      MOV #BOTTOM+4,R1
```



```

6293 032604 077104          SOB      R1,1$          ;OFF SO AS TO CREATE
6294 032606 052710 000076  BIS      #76,(R0)       ;THE 22-BIT PHYSICAL ADDRESS OF
6295                                     ;THE HIGHEST WORD IN
6296                                     ;MEMORY.
6297 032612 022020          CMP      (R0)+,(R0)+   ;DETERMINE THE RETURN ADDRESS
6298 032614 010066 000004  MOV      R0,4(SP)     ;AND LEAVE ON THE STACK FOR
6299                                     ;AN RTI.
6300 032620 012601          MOV      (SP)+,R1     ;RESTORE R1 AND R0.
6301 032622 012600          MOV      (SP)+,R0
6302 032624 000002          RTI
6303                                     ;RETURN
6304                                     ;THIS ROUTINE IS USED TO TYPE AN ERROR MESSAGE
6305                                     ;WHICH IS IN THE DATA TABLE. IT IS CALLED BY
6306                                     ;THE $ERROR ROUTINE OR BY FIRST SETTING THE $ITEMB
6307                                     ;BYTE EQUAL TO THE ERROR TABLE ITEM NUMBER THAT IS
6308                                     ;TO BE PRINTED OUT AND THEN EXECUTING A JSR PC,ERTYPE
6308 032626 104400          ERTYPE: TYPE
6309 032630 001305          .WORD  $CRLF

```

6310 032632 010046  
6311 032634 005000

MOV R0,-(SP) ;SAVE R0  
CLR R0



```

6312 032636 113700 001114      MOVB  $ITEMB,R0      ;GET THE ITEM NUMBER
6313 032642 001005              BNE   1$             ;ZERO?
6314 032644 013746 001116      MOV   $ERRPC,-(SP)   ;YES, TYPE JUST THE PC
6315 032650 104402              TYPOC                ;OF THE ERROR CALL.
6316 032652 000137 033170      JMP   ERT5
6317
6318 032656 005300              1$:  DEC   R0          ;MAKE R0 AN INDEX FOR THE
6319 032660 072027 000003      ASH   #3,R0         ;ERROR TABLE
6320 032664 062700 001314      ADD   # $ERRTB,R0
6321 032670 012037 032700      MOV   (R0)+,2$
6322 032674 001404              BEQ   3$             ;TYPE EM, ERROR MESSAGE.
6323 032676 104400              TYPE
6324 032700 000000              2$:  .WORD 0
6325 032702 104400              TYPE
6326 032704 001305              .WORD $CRLF
6327 032706 012037 032716      3$:  MOV   (R0)+,4$    ;TYPE DH, DATA HEADER
6328 032712 001404              BEQ   5$
6329 032714 104400              TYPE
6330 032716 000000              4$:  .WORD 0
6331 032720 104400              TYPE
6332 032722 001305              .WORD $CRLF
6333 032724 010146              5$:  MOV   R1,-(SP)    ;SAVE R1
6334 032726 012001              MOV   (R0)+,R1      ;GET DT, DATA TABLE ADDRESS
6335 032730 001002              BNE   6$
6336 032732 000137 033166      JMP   ERT4          ;JMP IF NO ERROR TABLE.
6337 032736 012000              6$:  MOV   (R0)+,R0    ;GET DF, DATA FORMAT ADDRESS
6338 032740 105710      ERT1: TSTB  (R0)        ;DATA FORMAT ENTRY EQUALS
6339 032742 001003              BNE   7$            ;ZERO?
6340 032744 013146              MOV   @ (R1)+,-(SP) ;YES, SO TYPE A 16-BIT
6341 032746 104402              TYPOC                ;OCTAL NUMBER
6342 032750 000500              BR    ERT2
6343 032752 122710 000001      7$:  CMPB  #1,(R0)     ;FORMAT EQUALS 1?
6344 032756 001003              BNE   8$
6345 032760 013146              MOV   @ (R1)+,-(SP) ;YES, TYPE A DECIMAL NUMBER
6346 032762 104410              TYPDS
6347 032764 000472              BR    ERT2
6348
6349 032766 122710 000002      8$:  CMPB  #2,(R0)     ;FORMAT 2?
6350 032772 001012              BNE   9$
6351 032774 012146              85$: MOV   (R1)+,-(SP) ;YES, TYPE A 22-BIT NUMBR
6352 032776 004737 031616      JSR   PC,$DB20      ;CALL $DB20 TO CONVERT THE
6353 033002 062716 000003      ADD   #3,(SP)       ;BINARY TO ASCII
6354 033006 012637 033014      MOV   (SP)+,29$     ;TYPE THE STRING
6355 033012 104400              TYPE
6356 033014 000000              29$: .WORD 0
6357 033016 000455              BR    ERT2
6358
6359 033020 122710 000004      9$:  CMPB  #4,(R0)     ;FORMAT 4?
6360 033024 001004              BNE  10$
6361 033026 013146              MOV   @ (R1)+,-(SP) ;YES, TYPE A 16-BIT
6362 033030 104404              TYPOS                ;OCTAL NUMBER SUPRESSING
6363 033032 016              .BYTE 16             ;LEADING ZEROES
6364 033033 000              .BYTE 0
6365 033034 000446              BR    ERT2
6366 033036 122710 000003      10$: CMPB  #3,(R0)     ;FORMAT 3?
6367 033042 001007              BNE  11$
    
```

```

6368 033044 013146          MOV      @ (R1)+, -(SP)      ;YES CONVER: 16-BIT
6369 033046 012737 177777 033174  MOV      #-1, TVADFL      ;VIRTUAL ADDRESS TO 32-BIT
6370 033054 004737 033202          JSR      PC, TYPVAD      ;PHYSICAL ADDRESS AND TYPE
6371 033060 000434          BR       ERT2           ;RELOCATE ONLY IF SEG. IS ON!
6372 033062 122710 000005 11$:  CMPB    #5, (R0)         ;FORMAT 5?
6373 033066 001005          BNE     12$
6374 033070 012137 033076          MOV      (R1)+, 20$      ;PRINT ASCII STRING
6375 033074 104400          TYPE
6376 033076 000000 20$:  .WORD  0
6377 033100 000426          BR       ERT3
6378
6379 033102 122710 000006 12$:  CMPB    #6, (R0)         ;FORMAT 6
6380 033106 001005          BNE     13$
6381 033110 005037 033174          CLR     TVADFL
6382 033114 004737 033202          JSR      PC, TYPVAD
6383 033120 000414          BR       ERT2
6384
6385 033122 122710 000007 13$:  CMPB    #7, (R0)         ;FORMAT 7?
6386 033126 001010          BNE     14$
6387 033130 012146          MOV      (R1)+, -(SP)
6388 033132 004737 031616          JSR      PC, $DB20
6389 033136 012637 033144          MOV      (SP)+, 45$
6390 033142 104400          TYPE
6391 033144 000000 45$:  .WORD  0
6392 033146 000401          BR       ERT2
6393
6394 033150 000000 14$:  HALT                    ;?????
6395
6396 033152 104400          ERT2:   TYPE
6397 033154 033450          .WORD  $TAB             ;PRINT A TAB AFTER TYPING AN
6398                                     ;ERROR TABLE ENTRY C ALL MODES
6399                                     ;EXCEPT ASCII
6400 ERT3:   INC      R0       ;POINT TO THE NEXT FORMAT BYTE
6401 033160 005711          TST     (R1)            ;IS THERE ANOTHER ENTRY?
6402 033162 001401          BEQ     ERT4
6403 033164 000665          BR       ERT1           ;YES, PROCESS IT
6404 033166 012601          ERT4:   MOV      (SP)+, R1 ;OTHERWISE:
6405 033170 012600          ERT5:   MOV      (SP)+, R0 ;RESTORE R1
6406 033172 000207          RTS     PC              ;RESTORE R0
6407                                     ;AND RETURN
6408 033174 000000          TVADFL: .WORD  0       ;FLAG USED TO TELL TYVAD
6409                                     ;WHETHER TO CONDITIONALLY
6410                                     ;OR UNCONDITIONALLY RELOCATE
6411                                     ;WHEN TYPING AN ADDRESS,
6412                                     ;-1 OR 0 RESPECTIVELY
6413
6414 033176 000000          TVADLO: .WORD  0       ;REGISTERS FOR THE 22-BIT
6415 033200 000000          TVADHI: .WORD  0       ;ADDRESS COMPUTED BY TYVAD.
6416
6417                                     ;ROUTINE WHICH CONVERTS A 16-BIT ADDRESS TO A 22-BIT
6418                                     ;ADDRESS. IF TVADFL IS -1, THEN CONVERT TO THE 22-BIT
6419                                     ;REAL ADDRESS DEPENDENT ON SEG BEING ON OR OFF FOR RELOCATION.
6420                                     ;IF TVADFL IS ZERO THEN UNCONDITIONAL USE THE KERNAL
6421                                     ;PAR WHICH IS APPROPRIATE TO DO RELOCATION.
6422 033202 104412          TYPVAD: SAVREG
6423 033204 016601 000002          MOV      2(SP), R1     ;GET THE VIRTUAL
    
```

```

6424 033210 010137 033176      MOV      R1,TVADLO      ;ADDRESS
6425 033214 005037 033200      CLR      TVADHI
6426 033220 005737 033174      TST      TVADFL      ;CONDITIONALLY RELOCATE?
6427 033224 001404      BEQ      1$
6428 033226 032737 000001 177572  BIT      #1,2$MMRO    ;YES, SEE IF MEMORY
6429 033234 001424      BEQ      2$           ;MANAGEMENT IS ON
6430 033236 005000      1$: CLR      R0        ;RELOCATE
6431 033240 073027 000003      ASHC     #3,R0       ;LEFT SHIFT R0 AND R1
6432 033244 006300      ASL      R0        ;THREE PLACES. R0 ONE
6433                                ;MORE SO THAT IT CONTAINS
6434                                ;2 X THE UPPER 3-BITS OF
6435 033246 000241      CLC
6436 033250 006001      ROR      R1        ;THE VIRTUAL ADDRESS
6437 033252 006001      ROR      R1        ;RESTORE R1 TO THE OFFSET
6438 033254 006001      ROR      R1        ;OF THE VIRTUAL ADDRESS
6439 033256 062700 172340  ADD      #KIPAR0,R0  ;TO THE PAR
6440                                ;DETERMINE THE CORRECT PAR'S
6441 033262 011003      MOV      (R0),R3    ;ADDRESS
6442 033264 005002      CLR      R2        ;GET ITS CONTENTS
6443 033266 073227 000006  ASHC     #6,R2       ;MAKE THE BLOCK COUNT
6444                                ;A 22-BIT ADDRESS.
6445 033272 060103      ADD      R1,R3     ;ADD THE OFFSET TO THE
6446 033274 005502      ADC      R2        ;BASE ADDRESS
6447
6448 033276 010237 033200      MOV      R2,TVADHI
6449 033302 010337 033176      MOV      R3,TVADLO
6450 033306 012746 033176      2$: MOV      #TVADLO,-(SP) ;CALL $DB20 TO CONVERT THE
6451 033312 004737 031616      JSR      PC,$DB20   ;22-BIT
6452 033316 062716 000003      ADD      #3,(SP)    ;TYPE ONLY 8 DIGITS.
6453 033322 012637 033330      MOV      (SP)+,3$
6454 033326 104400      TYPE
6455 033330 000000      3$: .WORD 0
6456 033332 104414      RESREG
6457 033334 012616      MOV      (SP)+,(SP) ;RESTORE THE REGISTERS
6458                                ;LEAVE ONLY THE RETURN
6459 033336 000207      RTS      PC        ;ADDRESS ON THE STACK.
6460                                ;RETURN
6461                                ;SPECIAL MESSAGES:
6462
6463 033340 041536 000200      CONCMS: .ASCIZ  '^C'<CRLF>
6464
6465 033344 047515 044516 047524  MMESRS: .ASCIZ  'MONITOR (OR LOADER) RESTORED!'<CRLF>
6466 033352 020122 047450 020122
6467 033360 047514 042101 051105
6468 033366 020051 042522 052123
6469 033374 051117 042105 100041
6470 033402      000
6471
6472 033403      200 047520 042527  POWERM: .ASCIZ  <CRLF>'POWER FAILURE, PROGRAM RESTARTING'<CRLF><CRLF>
6473 033410 020122 040506 046111
6474 033416 051125 026105 050040
6475 033424 047522 051107 046501
6476 033432 051040 051505 040524
6477 033440 052122 047111 100107
6478 033446 000200
6479

```

6480	033450	000011			\$TAB:	.ASCIZ	<TAB>
6481							
6482	033452	042600	050130	041505	MTA5:	.ASCII	<CRLF>'EXPECTED DATA:'<CRLF>
6483	033460	042524	020104	040504			
6484	033466	040524	100072				
6485	033472	051107	052517	020120		.ASCIZ	'GROUP 0.GROUP 1.MEM EV.'<TAB>'MEM ODD.'<CRLF>
6486	033500	027060	051107	052517			
6487	033506	020120	027061	042515			
6488	033514	020115	053105	004456			
6489	033522	042515	020115	042117			
6490	033530	027104	000200				
6491							
6492	033534	042200	052101	020101	MTA17:	.ASCII	<CRLF>'DATA WRITTEN.'<TAB>'TEST ADDR.'<TAB>'ERROR REG.'<CRLF>
6493	033542	051127	052111	042524			
6494	033550	027116	052011	051505			
6495	033556	020124	042101	051104			
6496	033564	004456	051105	047522			
6497	033572	020122	042522	027107			
6498	033600	200					
6499							
6500	033601	040	047111	000040	MTA17:	.ASCIZ	' IN '
6501							
6502	033606	054105	042520	052103	MTB17:	.ASCIZ	'EXPECTED DATA:'<CRLF>
6503	033614	042105	042040	052101			
6504	033622	035101	000200				
6505							
6506	033626	054502	042524	004456	MTC17:	.ASCIZ	'BYTE.'<TAB>
6507	033634	000					
6508							
6509	033635	127	051117	027104	MTA20:	.ASCIZ	'WORD.'<TAB>
6510	033642	000011					
6511							
6512	033644	054105	042520	052103	MTA21:	.ASCII	'EXPECTED DATA:'<CRLF>
6513	033652	042105	042040	052101			
6514	033660	035101	200				
6515	033663	110	052111	020123		.ASCIZ	'HITS IN GROUP 0.'<TAB>'/'<TAB>'HITS IN GROUP 1. '<CRLF>
6516	033670	047111	043440	047522			
6517	033676	050125	030040	004456			
6518	033704	004457	044510	051524			
6519	033712	044440	020116	051107			
6520	033720	052517	020120	027061			
6521	033726	100040	000				
6522							
6523		033601			MTB21-MTA17		
6524							
6525	033731	200	042524	052123	MTA43:	.ASCII	<CRLF>'TEST ADDRESS.'<TAB>'ERROR ADRS REG.'<TAB>
6526	033736	040440	042104	042522			
6527	033744	051523	004456	051105			
6528	033752	047522	020122	042101			
6529	033760	051522	051040	043505			
6530	033766	004456					
6531	033770	051105	047522	020122		.ASCIZ	'ERROR REG.'<CRLF>
6532	033776	042522	027107	000200			
6533							
6534	034004	053600	047522	042524	MTA45:	.ASCIZ	<CRLF>'WROTE. 377'<TAB>'IN BYTE. '
6535	034012	020056	033463	004467			

6536	034020	047111	041040	052131	
6537	034026	027105	000040		
6538					
6539	034032	051200	040505	020104	MTB45: .ASCIIZ <CRLF>'READ DATA. '
6540	034040	040504	040524	020056	
6541	034046	000			
6542					
6543	034047	011	047111	053440	MTC45: .ASCIIZ <TAB>'IN WORD. '
6544	034054	051117	027104	000040	
6545					
6546	034062	053600	047522	042524	MTA50: .ASCIIZ <CRLF>'WROTE. 000'<TAB>'IN BYTE. '
6547	034070	020056	030060	004460	
6548	034076	047111	041040	052131	
6549	034104	027105	000040		
6550					
6551	034110	042600	052116	051105	PDMSG1: .ASCII <CRLF>'ENTERING CACHE ADDRESS MEMORY POWER UP '
6552	034116	047111	020107	040503	
6553	034124	044103	020105	042101	
6554	034132	051104	051505	020123	.
6555	034140	042515	047515	054522	.
6556	034146	050040	053517	051105	.
6557	034154	052440	020120		
6558	034160	047111	040526	044514	.ASCII 'INVALIDATOR TEST.'<CRLF>
6559	034166	040504	047524	020122	
6560	034174	042524	052123	100056	
6561	034202	046120	040505	042523	.ASCII 'PLEASE GO THROUGH A POWER DOWN, POWER UP '
6562	034210	043440	020117	044124	
6563	034216	047522	043525	020110	
6564	034224	020101	047520	042527	
6565	034232	020122	047504	047127	
6566	034240	020054	047520	042527	
6567	034246	020122	050125	040	
6568	034253	123	050505	042525	.ASCIIZ 'SEQUENCE.'<CRLF>
6569	034260	041516	027105	000200	
6570					
6571	034266	041600	041501	042510	PDMSG2: .ASCII <CRLF>'CACHE ADDRESS MEMORY POWER UP INVALIDATOR'
6572	034274	040440	042104	042522	
6573	034302	051523	046440	046505	
6574	034310	051117	020131	047520	
6575	034316	042527	020122	050125	
6576	034324	044440	053116	046101	
6577	034332	042111	052101	051117	
6578	034340	052040	051505	020124	.ASCIIZ ' TEST DID NOT FAIL.'<CRLF>
6579	034346	044504	020104	047516	
6580	034354	020124	040506	046111	
6581	034362	100056	000		
6582					
6583	034365	105	051122	051117	ADRNG: .ASCII 'ERROR ADDRESS REGISTER NEEDED FOR TEST,'<CRIF>'BUT IT HAS BEEN '
6584	034372	040440	042104	042522	
6585	034400	051523	051040	043505	
6586	034406	051511	042524	020122	
6587	034414	042516	042105	042105	
6588	034422	043040	051117	052040	
6589	034430	051505	026124	041200	
6590	034436	052125	044440	020124	
6591	034444	040510	020123	042502	

6592	034452	047105	040		
6593	034455	106	040514	043507	.ASCIZ 'FLAGGED AS BAD.'
6594	034462	042105	040440	020123	
6595	034470	040502	020504	000	
6596					
6597	034475	105	051122	051117	ERRNG: .ASCII 'ERROR REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6598	034502	051040	043505	051511	
6599	034510	042524	020122	042516	
6600	034516	042105	042105	043040	
6601	034524	051117	052040	051505	
6602	034532	026124	041200	052125	
6603	034540	044440	020124	040510	
6604	034546	020123	042502	047105	
6605	034554	040			
6606	034555	106	040514	043507	.ASCIZ 'FLAGGED AS BAD.'
6607	034562	042105	040440	020123	
6608	034570	040502	020504	000	
6609					
6610	034575	103	047117	051124	CNRNG: .ASCII 'CONTRCL REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6611	034602	046117	051040	043505	
6612	034610	051511	042524	020122	
6613	034616	042516	042105	042105	
6614	034624	043040	051117	052040	
6615	034632	051505	026124	041200	
6616	034640	052125	044440	020124	
6617	034646	040510	020123	042502	
6618	034654	047105	040		
6619	034657	106	040514	043507	.ASCIZ 'FLAGGED AS BAD.'
6620	034664	042105	040440	020123	
6621	034672	040502	020504	000	
6622	034677	115	044501	052116	MNRNG: .ASCII 'MAINTENANCE REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6623	034704	047105	047101	042503	
6624	034712	051040	043505	051511	
6625	034720	042524	020122	042516	
6626	034726	042105	042105	043040	
6627	034734	051117	052040	051505	
6628	034742	026124	041200	052125	
6629	034750	044440	020124	040510	
6630	034756	020123	042502	047105	
6631	034764	040			
6632	034765	106	040514	043507	.ASCIZ 'FLAGGED AS BAD.'
6633	034772	042105	040440	020123	
6634	035000	040502	020504	000	
6635					
6636	035005	110	052111	046457	HMRNG: .ASCII 'HIT/MISS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6637	035012	051511	020123	042522	
6638	035020	044507	052123	051105	
6639	035026	047040	042505	042504	
6640	035034	020104	047506	020122	
6641	035042	042524	052123	100054	
6642	035050	052502	020124	052111	
6643	035056	044040	051501	041040	
6644	035064	042505	020116		
6645	035070	046106	043501	042507	.ASCIZ 'FLAGGED AS BAD.'
6646	035076	020104	051501	041040	
6647	035104	042101	000041		

```

6648
6649 035110 040600 042104 042522 MTA77: .ASCIIZ <CRLF>'ADDRESS: '
6650 035116 051523 020072 000040
6651
6652 035124 051440 047510 046125 MTB77: .ASCIIZ ' SHOULD HAVE BEEN A HIT IN GROUP '
6653 035132 020104 040510 042526
6654 035140 041040 042505 020116
6655 035146 020101 044510 020124
6656 035154 047111 043440 047522
6657 035162 050125 000040
6658
6659 035166 043101 042524 020122 MTC77: .ASCIIZ 'AFTER REFERENCING'<CRLF>'ADDRESS: '
6660 035174 042522 042506 042522
6661 035202 041516 047111 100107
6662 035210 042101 051104 051505
6663 035216 035123 020040 000
6664
6665 035223 040 044127 046111 MTD77: .ASCIIZ ' WHILE FORCING SELECTION OF GROUP '
6666 035230 020105 047506 041522
6667 035236 047111 020107 042523
6668 035244 042514 052103 047511
6669 035252 020116 043117 043440
6670 035260 047522 050125 000040
6671
6672 035266 040600 051122 051117 MTA101: .ASCII <CRLF>'ERROR ADRS REG.'<TAB>'ERROR REG.'<TAB>
6673 035274 040440 051104 020123
6674 035302 042522 027107 042411
6675 035310 051122 051117 051040
6676 035316 043505 004456
6677 035322 054105 042520 052103 .ASCIIZ 'EXPECTED ERR.'<TAB>'PATTERN PUT IN MAINT REG.'<CRLF>
6678 035330 042105 042440 051122
6679 035336 004456 040520 052124
6680 035344 051105 020116 052520
6681 035352 020124 047111 046440
6682 035360 044501 052116 051040
6683 035366 043505 100056 000
6684
6685 035373 200 043101 042524 MTA120: .ASCIIZ <CRLF>'AFTER 2ND CYCLE READ '
6686 035400 020122 047062 020104
6687 035406 054503 046103 020105
6688 035414 042522 042101 020040
6689 035422 000
6690
6691 035423 200 043101 042524 MTB120: .ASCIIZ <CRLF>'AFTER 4TH CYCLE READ '
6692 035430 020122 052064 020110
6693 035436 054503 046103 020105
6694 035444 042522 042101 020040
6695 035452 000
6696
6697 035453 200 043101 042524 MTC120: .ASCIIZ <CRLF>'AFTER 6TH CYCLE READ '
6698 035460 020122 052066 020110
6699 035466 054503 046103 020105
6700 035474 042522 042101 020040
6701 035502 000
6702 035503 200 043101 042524 MTD120: .ASCIIZ <CRLF>'AFTER 8TH CYCLE READ '
6703 035510 020122 052070 020110

```

6704	035516	054503	046103	020105	
6705	035524	042522	042101	020040	
6706	035532	000			
6707					
6708	035533	200	043101	042524	MTE120: .ASCIIZ <CRLF>'AFTER 10TH CYCLE READ '
6709	035540	020122	030061	044124	
6710	035546	041440	041531	042514	
6711	035554	051040	040505	020104	
6712	035562	000			
6713					
6714	035563	200	043101	042524	MTF120: .ASCIIZ <CRLF>'AFTER 12TH CYCLE READ '
6715	035570	020122	030061	044124	
6716	035576	041440	041531	042514	
6717	035604	051040	040505	020104	
6718	035612	000			
6719					
6720	035613	106	047522	020115	MTG120: .ASCIIZ 'FROM THE HIT/MISS REG. EXPECTED '
6721	035620	044124	020105	044510	
6722	035626	027524	044515	051523	
6723	035634	051040	043505	020056	
6724	035642	054105	042520	052103	
6725	035650	042105	000040		
6726					
6727	035654	052200	042510	050040	MTA124: .ASCII <CRLF>'THE PATTERN BEING USED IN THE MAINTENANCE '
6728	035662	052101	042524	047122	
6729	035670	041040	044505	043516	
6730	035676	052440	042523	020104	
6731	035704	047111	052040	042510	
6732	035712	046440	044501	052116	
6733	035720	047105	047101	042503	
6734	035726	040			
6735	035727	122	043505	051511	.ASCIIZ 'REGISTER WAS: '
6736	035734	042524	020122	040527	
6737	035742	035123	000040		
6738					
6739	035746	051200	043105	051105	MTA126: .ASCIIZ <CRLF>'REFERENCED ADDRESS:'<TAB>
6740	035754	047105	042503	020104	
6741	035762	042101	051104	051505	
6742	035770	035123	000011		
6743					
6744	035774	040600	051122	051117	MTB126: .ASCIIZ <CRLF>'ERROR ADDRESS REGISTER:'<TAB>
6745	036002	040440	042104	042522	
6746	036010	051523	051040	043505	
6747	036016	051511	042524	035122	
6748	036024	000011			
6749					
6750	036026	050200	052101	042524	MTA131: .ASCIIZ <CRLF>'PATTERN BEING USED IN THE MAINTENANCE REGISTER:'<TAB>
6751	036034	047122	041040	044505	
6752	036042	043516	052440	042523	
6753	036050	020104	047111	052040	
6754	036056	042510	046440	044505	
6755	036064	052116	047105	047101	
6756	036072	042503	051040	043505	
6757	036100	051511	042524	035122	
6758	036106	000011			
6759					



6760	036110	042600	050130	041505	MTB131: .ASCIIZ <CRLF>'EXPECTED ERROR REGISTER:'<TAB>
6761	036116	042524	020104	051105	
6762	036124	047522	020122	042522	
6763	036132	044507	052123	051105	
6764	036140	004472	000		
6765					
6766	036143	200	047507	020124	MTC131: .ASCIIZ <CRLF>'GOT ERROR REGISTER:'<TAB>
6767	036150	051105	047522	020122	
6768	036156	042522	044507	052123	
6769	036164	051105	004472	000	
6770					
6771	036171	200	051105	047522	MTA134: .ASCIIZ <CRLF>'ERROR ADR REG.'<TAB>'ERROR REG.'<CRLF>
6772	036176	020122	042101	020122	
6773	036204	042522	027107	042411	
6774	036212	051122	051117	051040	
6775	036220	043505	100056	000	
6776					
6777	036225	200	054105	042520	MTA135: .ASCIIZ <CRLF>'EXPECTED ERROR REG.: '
6778	036232	052103	042105	042440	
6779	036240	051122	051117	051040	
6780	036246	043505	035056	020040	
6781	036254	000			
6782					
6783	036255	107	052117	042440	MTB135: .ASCIIZ 'GOT ERROR REG.: '
6784	036262	051122	051117	051040	
6785	036270	043505	035056	020040	
6786	036276	000			
6787					
6788	036277	200	054105	042520	MTC135: .ASCIIZ <CRLF>'EXPECTED ERROR ADR REG.: '
6789	036304	052103	042105	042440	
6790	036312	051122	051117	040440	
6791	036320	051104	051040	043505	
6792	036326	035056	020040	000	
6793					
6794	036333	107	052117	042440	MTD135: .ASCIIZ 'GOT ERROR ADR REG.: '
6795	036340	051122	051117	040440	
6796	036346	051104	051040	043505	
6797	036354	035056	020040	000	
6798	036361	200	050103	020125	MSG1: .ASCIIZ<CRLF> ''CPU UNDER TEST FOUND TO BE A ''
6799	036366	047125	042504	020122	
6800	036374	042524	052123	043040	
6801	036402	052517	042116	052040	
6802	036410	020117	042502	040440	
6803	036416	000040			
6804	036420	041113	030461	042455	MSG2: .ASCIIZ 'KB11-EM'<CRLF>
6805	036426	100115	000		
6806	036431	113	030502	026461	MSG3: .ASCIIZ 'KB11-B/C'<CRLF>
6807	036436	027502	100103	000	
6808	036443	113	030502	026461	MSG4: .ASCIIZ 'KB11-CM ''<CRLF>
6809	036450	046503	020040	020040	
6810	036456	020040	020040	020040	
6811	036464	020040	020040	020040	
6812	036472	000200			
6813	036474	041113	030461	042455	MSG5: .ASCIIZ 'KB11-E'<CRLF>
6814	036502	000200			
6815					

```
6816                                     : THESE ARE THE ERROR MESSAGES:
6817
6818 036504 020101 042522 042506 EM1: .ASCIZ 'A REFERENCE WHICH SHOULD HAVE BEEN A HIT WAS A MISS.'
6819 036512 042522 041516 020105
6820 036520 044127 041511 020110
6821 036526 044123 052517 042114
6822 036534 044040 053101 020105
6823 036542 042502 047105 040440
6824 036550 044040 052111 053440
6825 036556 051501 040440 046440
6826 036564 051511 027123 000
6827
6828
6829 036571 200 047125 054105 EM14: .ASCIZ <CRLF>'UNEXPECTED PARITY ERROR TRAP.'
6830 036576 042520 052103 042105
6831 036604 050040 051101 052111
6832 036612 020131 051105 047522
6833 036620 020122 051124 050101
6834 036626 000056
6835
6836 036630 025052 052052 051505 EM15: .ASCIZ '***TEST ABORTED! GOING TO NEXT TEST.***'
6837 036636 020124 041101 051117
6838 036644 042524 020504 043440
6839 036652 044517 043516 052040
6840 036660 020117 042516 052130
6841 036666 052040 051505 027124
6842 036674 025052 000052
6843 036700 040503 044103 020105 EM55: .ASCII 'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
6844 036706 042522 044507 052123
6845 036714 051105 051040 051505
6846 036722 047520 051516 020105
6847 036730 042524 052123 043040
6848 036736 044501 042514 027104
6849 036744 200
6850 036745 101 051040 043105 .ASCII 'A REFERENCE TO THE LOW ORDER ERROR ADDRESS REGISTER '
6851 036752 051105 047105 042503
6852 036760 052040 020117 044124
6853 036766 020105 047514 020127
6854 036774 051117 042504 020122
6855 037002 051105 047522 020122
6856 037010 042101 051104 051505
6857 037016 020123 042522 044507
6858 037024 052123 051105 040
6859 037031 124 046511 042105 .ASCIZ 'TIMED OUT.'
6860 037036 047440 052125 000056
6861
6862 037044 040503 044103 020105 EM56: .ASCII 'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
6863 037052 042522 044507 052123
6864 037060 051105 051040 051505
6865 037066 047520 051516 020105
6866 037074 042524 052123 043040
6867 037102 044501 042514 027104
6868 037110 200
6869 037111 101 051040 043105 .ASCII 'A REFERENCE TO THE HIGH ORDER ERROR ADDRESS REGISTER '
6870 037116 051105 047105 042503
6871 037124 052040 020117 044124
```

6872	037132	020105	044510	044107	
6873	037140	047440	042122	051105	
6874	037146	042440	051122	051117	
6875	037154	040440	042104	042522	
6876	037162	051523	051040	043505	
6877	037170	051511	042524	020122	
6878	037176	044524	042515	020104	.ASCIZ 'TIMED OUT.'
6879	037204	052517	027124	000	
6880					
6881	037211	103	041501	042510	EM57: .ASCII 'CACHE REGISTER RESPONSE TEST FAILED.<CRLF>
6882	037216	051040	043505	051511	
6883	037224	042524	020122	042522	
6884	037232	050123	047117	042523	
6885	037240	052040	051505	020124	
6886	037246	040506	046111	042105	
6887	037254	100056			
6888	037256	020101	042522	042506	.ASCIZ 'A REFERENCE TO THE ERROR REGISTER TIMED OUT.'
6889	037264	042522	041516	020105	
6890	037272	047524	052040	042510	
6891	037300	042440	051122	051117	
6892	037306	051040	043505	051511	
6893	037314	042524	020122	044524	
6894	037322	042515	020104	052517	
6895	037330	027124	000		
6896					
6897	037333	103	041501	042510	EM60: .ASCII 'CACHE REGISTER RESPONSE TEST FAILED.<CRLF>
6898	037340	051040	043505	051511	
6899	037346	042524	020122	042522	
6900	037354	050123	047117	042523	
6901	037362	052040	051505	020124	
6902	037370	040506	046111	042105	
6903	037376	100056			
6904	037400	020101	042522	042506	.ASCIZ 'A REFERENCE TO THE CONTROL REGISTER TIMED OUT.'
6905	037406	042522	041516	020105	
6906	037414	047524	052040	042510	
6907	037422	041440	047117	051124	
6908	037430	046117	051040	043505	
6909	037436	051511	042524	020122	
6910	037444	044524	042515	020104	
6911	037452	052517	027124	000	
6912					
6913	037457	103	041501	042510	EM61: .ASCII 'CACHE REGISTER RESPONSE TEST FAILED.<CRLF>
6914	037464	051040	043505	051511	
6915	037472	042524	020122	042522	
6916	037500	050123	047117	042523	
6917	037506	052040	051505	020124	
6918	037514	040506	046111	042105	
6919	037522	100056			
6920	037524	020101	042522	042506	.ASCIZ 'A REFERENCE TO THE MAINTENANCE REGISTER TIMED OUT.'
6921	037532	042522	041516	020105	
6922	037540	047524	052040	042510	
6923	037546	046440	044501	052116	
6924	037554	047105	047101	042503	
6925	037562	051040	043505	051511	
6926	037570	042524	020122	044524	
6927	037576	042515	020104	052517	

6928	037604	027124	000		
6929					
6930	037607	103	041501	042510	EM62: .ASCII 'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
6931	037614	051040	043505	051511	
6932	037622	042524	020122	042522	
6933	037630	050123	047117	042523	
6934	037636	052040	051505	020124	
6935	037644	040506	046111	042105	
6936	037652	100056			
6937	037654	020101	042522	042506	.ASCIIZ 'A REFERENCE TO THE HIT/MISS REGISTER TIMED OUT.'<CRLF>
6938	037662	042522	041516	020105	
6939	037670	047524	052040	042510	
6940	037676	044040	052111	046457	
6941	037704	051511	020123	042522	
6942	037712	044507	052123	051105	
6943	037720	052040	046511	042105	
6944	037726	047440	052125	100056	
6945	037734	000			
6946					
6947	037735	103	041501	042510	EM63: .ASCII 'CACHE REGISTER DATA PATHS, READ ZEROES, TEST FAILED.'
6948	037742	051040	043505	051511	
6949	037750	042524	020122	040504	
6950	037756	040524	050040	052101	
6951	037764	051510	020054	042522	
6952	037772	042101	055040	051105	
6953	040000	042517	026123	052040	
6954	040006	051505	020124	040506	
6955	040014	046111	042105	056	
6956	040021	200	051127	052117	.ASCII <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA '
6957	040026	020105	042532	047522	
6958	040034	051505	041040	052125	
6959	040042	051040	040505	020104	
6960	040050	040502	045503	047040	
6961	040056	047117	055055	051105	
6962	040064	020117	040504	040524	
6963	040072	040			
6964	040073	106	047522	020115	.ASCIIZ 'FROM BOTH'<CRLF>'THE CONTROL AND MAINTENANCE REGISTERS.'
6965	040100	047502	044124	052200	
6966	040106	042510	041440	047117	
6967	040114	051124	046117	040440	
6968	040122	042116	046440	04450'	
6969	040130	052116	047105	047101	
6970	040136	042503	051040	043505	
6971	040144	051511	042524	051522	
6972	040152	000056			
6973					
6974	040154	040503	044103	020105	EM64: .ASCII 'CACHE REGISTER DATA PATH, READ ZEROES, TEST FAILED.'
6975	040162	042522	044507	052123	
6976	040170	051105	042040	052101	
6977	040176	020101	040520	044124	
6978	040204	020054	042522	042101	
6979	040212	055040	051105	042517	
6980	040220	026123	052040	051505	
6981	040226	020124	040506	046111	
6982	040234	042105	056		
6983	040237	200	051127	052117	.ASCII <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA FROM '

```

6984 040244 020105 042532 047522
6985 040252 051505 041040 052125
6986 040260 051040 040505 020104
6987 040266 040502 045503 047040
6988 040274 047117 055055 051105
6989 040302 020117 040504 040524
6990 040310 043040 047522 020115
6991 040316 052200 042510 041440
6992 040324 041501 042510 041440
6993 040332 047117 051124 046117
6994 040340 051040 043505 051511
6995 040346 042524 027122 000
6996
6997 040353 103 041501 042510
6998 040360 051040 043505 051511
6999 040366 042524 020122 040504
7000 040374 040524 050040 052101
7001 040402 051510 020054 042522
7002 040410 042101 047440 042516
7003 040416 026123 051040 051505
7004 040424 020124 040506 046111
7005 040432 042105 100056
7006 040436 040506 046111 042105
7007 040444 052040 020117 042522
7008 040452 042101 041440 051117
7009 040460 042522 052103 042040
7010 040466 052101 020101 051106
7011 040474 046517 052040 042510
7012 040502 040440 042104 042522
7013 040510 051523 051040 043505
7014 040516 051511 042524 122
7015 040523 040 047111 052040
7016 040530 042510 041440 042514
7017 040536 051101 051440 040524
7018 040544 042524 100056 044124
7019 040552 020105 047514 020127
7020 040560 051117 042504 020122
7021 040566 042101 051104 051505
7022 040574 020123
7023 040576 044123 052517 042114
7024 040604 044040 053101 020105
7025 040612 042502 047105 051440
7026 040620 052105 052040 035117
7027 040626 030440 033467 032067
7028 040634 100060
7029 040636 044124 020105 044510
7030 040644 044107 047440 042122
7031 040652 051105 040440 042104
7032 040660 042522 051523 051040
7033 040666 043505 051511 042524
7034 040674 020122 044123 052517
7035 040702 042114 044040 053101
7036 040710 020105 042502 047105
7037 040716 040
7038 040717 123 052105 052040
7039 040724 035117 030040 030060

```

.ASCIZ <CRLF>'THE CACHE CONTROL REGISTER.'

EM65: .ASCII 'CACHE REGISTER DATA PATHS, READ ONES, REST FAILED.'<CRLF>

.ASCII 'FAILED TO READ CORRECT DATA FROM THE ADDRESS REGISTER'

.ASCII ' IN THE CLEAR STATE.'<CRLF>'THE LOW ORDER ADDRESS '

.ASCII 'SHOULD HAVE BEEN SET TO: 177740'<CRLF>

.ASCII 'THE HIGH ORDER ADDRESS REGISTER SHOULD HAVE BEEN '

.ASCIZ 'SET TO: 000003'

7040	040732	030060	000063		
7041					
7042	040736	040503	044103	020105	EM66: .ASCIIZ 'CACHE CONTROL REGISTER COUNT PATTERN TEST FAILED.'
7043	040744	047503	052116	047522	
7044	040752	020114	042522	044507	
7045	040760	052123	051105	041440	
7046	040766	052517	052116	050040	
7047	040774	052101	042524	047122	
7048	041002	052040	051505	020124	
7049	041010	040506	046111	042105	
7050	041016	000056			
7051					
7052	041020	040503	044103	020105	EM67: .ASCII 'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
7053	041026	044510	027524	044515	
7054	041034	051523	040440	042116	
7055	041042	041440	047117	051124	
7056	041050	046117	051040	043505	
7057	041056	051511	042524	020122	
7058	041064	042524	052123	043040	
7059	041072	044501	042514	027104	
7060	041100	053600	052111	020110	.ASCII <CR,F>'WITH THE CONTROL REGISTER CLEAR, THE HIT/MISS '
7061	041106	044124	020105	047503	
7062	041114	052116	047522	020114	
7063	041122	042522	044507	052123	
7064	041130	051105	041440	042514	
7065	041136	051101	020054	044124	
7066	041144	020105	044510	027524	
7067	041152	044515	051523	040	
7068	041157	122	043505	051511	.ASCIIZ 'REGISTER SHOULD'<CRLF>'HAVE SHOWN SIX HITS (000077).'
7069	041164	042524	020122	044123	
7070	041172	052517	042114	044200	
7071	041200	053101	020105	044123	
7072	041206	053517	020116	044523	
7073	041214	020130	044510	051524	
7074	041222	024040	030060	030060	
7075	041230	033467	027051	000	
7076					
7077	041235	103	041501	042510	EM70: .ASCII 'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
7078	041242	044040	052111	046457	
7079	041250	051511	020123	047101	
7080	041256	020104	047503	052116	
7081	041264	047522	020114	042522	
7082	041272	044507	052123	051105	
7083	041300	052040	051505	020124	
7084	041306	040506	046111	042105	
7085	041314	056			
7086	041315	200	044127	046111	.ASCII <CRLF>'WHILE FORCING SELECTION OF GROUP 1 AND FORCING '
7087	041322	020105	047506	041522	
7088	041330	047111	020107	042523	
7089	041336	042514	052103	047511	
7090	041344	020116	043117	043440	
7091	041352	047522	050125	030440	
7092	041360	040440	042116	043040	
7093	041366	051117	044503	043516	
7094	041374	040			
7095	041375	115	051511	042523	.ASCII 'MISSES TO GROUP 0,'<CRLF>'THE HIT/MISS REGISTER '

7096	041402	020123	047524	043440
7097	041410	047522	050125	030040
7098	041416	100054	044124	020105
7099	041424	044510	027524	044515
7100	041432	051523	051040	043505
7101	041440	051511	042524	020122
7102	041446	044123	052517	042114
7103	041454	044040	053101	020105
7104	041462	044123	053517	020116
7105	041470	044523	020130	044510
7106	041476	051524	024040	030060
7107	041504	030060	033467	027051
7108	041512	000		
7109				
7110	041513	103	041501	042510
7111	041520	044040	052111	046457
7112	041526	051511	020123	047101
7113	041534	020104	047503	052116
7114	041542	047522	020114	042522
7115	041550	044507	052123	051105
7116	041556	052040	051505	020124
7117	041564	040506	046111	042105
7118	041572	056		
7119	041573	200	044127	046111
7120	041600	020105	047506	041522
7121	041606	047111	020107	042523
7122	041614	042514	052103	047511
7123	041622	020116	043117	043440
7124	041630	047522	050125	030040
7125	041636	040440	042116	043040
7126	041644	051117	044503	043516
7127	041652	040		
7128	041653	115	051511	042523
7129	041660	020123	047524	043440
7130	041666	047522	050125	030440
7131	041674	100054	044124	020105
7132	041702	044510	027524	044515
7133	041710	051523	051040	043505
7134	041716	051511	042524	020122
7135	041724	044123	052517	042114
7136	041732	044040	053101	020105
7137	041740	044123	053517	020116
7138	041746	044523	020130	044510
7139	041754	051524	024040	030060
7140	041762	030060	033467	027051
7141	041770	000		
7142				
7143	041771	103	041501	042510
7144	041776	044040	052111	046457
7145	042004	051511	020123	047101
7146	042012	020104	047503	052116
7147	042020	047522	020114	042522
7148	042026	044507	052123	051105
7149	042034	052040	051505	020124
7150	042042	040506	046111	042105
7151	042050	056		

.ASCIZ 'SHOULD HAVE SHOWN SIX HITS (000077).'

EM71: .ASCII 'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'

.ASCII <CRLF>'WHILE FORCING SELECTION OF GROUP 0 AND FORCING '

.ASCII 'MISSES TO GROUP 1,'<CRLF>'THE HIT/MISS REGISTER '

.ASCIZ 'SHOULD HAVE SHOWN SIX HITS (000077).'

EM72: .ASCII 'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'

7152	042051	127	044510	042514		.ASCII 'WHILE FORCING MISSES TO BOTH GROUPS, THE HIT/MISS '
7153	042056	043040	051117	044503		
7154	042064	043516	046440	051511		
7155	042072	042523	020123	047524		
7156	042100	041040	052117	020110		
7157	042106	051107	052517	051520		
7158	042114	020054	044124	020105		
7159	042122	044510	027524	044515		
7160	042130	051523	040			
7161	042133	122	043505	051511		.ASCIIZ 'REGISTER'<CRLF>'SHOULD HAVE SHOWN SIX MISSES (000000).'
7162	042140	042524	100122	044123		
7163	042146	052517	042114	044040		
7164	042154	053101	020105	044123		
7165	042162	053517	020116	044523		
7166	042170	020130	044515	051523		
7167	042176	051505	024040	030060		
7168	042204	030060	030060	027051		
7169	042212	000				
7170						
7171	042213	103	041501	042510	EM73:	.ASCII 'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
7172	042220	044040	052111	046457		
7173	042226	051511	020123	047101		
7174	042234	020104	047503	052116		
7175	042242	047522	020114	042522		
7176	042250	044507	052123	051105		
7177	042256	052040	051505	020124		
7178	042264	040506	046111	042105		
7179	042272	056				
7180	042273	200	044127	046111		.ASCII <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '
7181	042300	020105	047506	041522		
7182	042306	047111	020107	044515		
7183	042314	051523	051505	052040		
7184	042322	020117	047502	044124		
7185	042330	043440	047522	050125		
7186	042336	020123	047101	020104		
7187	042344	047506	041522	047111		
7188	042352	020107				
7189	042354	042523	042514	052103		.ASCII 'SELECTION OF GROUP 1,'<CRLF>'THE HIT/MISS REGISTER '
7190	042362	047511	020116	043117		
7191	042370	043440	047522	050125		
7192	042376	030440	100054	044124		
7193	042404	020105	044510	027524		
7194	042412	044515	051523	051040		
7195	042420	043505	051511	042524		
7196	042426	020122				
7197	042430	044123	052517	042114		.ASCIIZ 'SHOULD HAVE SHOWN SIX MISSES (000000).'
7198	042436	044040	053101	020105		
7199	042444	044123	053517	020116		
7200	042452	044523	020130	044515		
7201	042460	051523	051505	024040		
7202	042466	030060	030060	030060		
7203	042474	027051	000			
7204						
7205	042477	103	041501	042510	EM74:	.ASCII 'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
7206	042504	044040	052111	046457		
7207	042512	051511	020123	047101		



7208	042520	020104	047503	052116
7209	042526	047522	020114	042522
7210	042534	044507	052123	051105
7211	042542	052040	051505	020124
7212	042550	040506	046111	042105
7213	042556	056		
7214	042557	200	044127	046111
7215	042564	020105	047506	041522
7216	042572	047111	020107	044515
7217	042600	051523	051505	052040
7218	042606	020117	047502	044124
7219	042614	043440	047522	050125
7220	042622	020123	047101	020104
7221	042630	047506	041522	047111
7222	042636	020107		
7223	042640	042523	042514	052103
7224	042646	047511	020116	043117
7225	042654	043440	047522	050125
7226	042662	030040	100054	044124
7227	042670	020105	044510	027524
7228	042676	044515	051523	051040
7229	042704	043505	051511	042524
7230	042712	020122		
7231	042714	044123	052517	042114
7232	042722	044040	053101	020105
7233	042730	044123	053517	020116
7234	042736	044523	020130	044515
7235	042744	051523	051505	024040
7236	042752	030060	030060	030060
7237	042760	027051	000	
7238				
7239	042763	103	047117	051124
7240	042770	046117	051040	043505
7241	042776	051511	042524	020122
7242	043004	042524	052123	043040
7243	043012	044501	042514	027104
7244	043020	043200	044501	042514
7245	043026	020104	047524	043440
7246	043034	052105	040	
7247	043037	101	044040	052111
7248	043044	047440	020116	020101
7249	043052	042522	042506	042522
7250	043060	041516	020105	044127
7251	043066	041511	020110	044123
7252	043074	052517	042114	044040
7253	043102	053101	020105	042502
7254	043110	047105	040440	044040
7255	043116	052111	000056	
7256				
7257		042763		
7258				
7259	043122	047503	052116	047522
7260	043130	020114	042522	044507
7261	043136	052123	051105	052040
7262	043144	051505	020124	040506
7263	043152	046111	042105	100056

.ASCII <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '

.ASCII 'SELECTION OF GROUP 0,'<CRLF>'THE HIT/MISS REGISTER '

.ASCIIZ 'SHOULD HAVE SHOWN SIX MISSES (000000).'

EM75: .ASCII 'CONTROL REGISTER TEST FAILED.'<CRLF>'FAILED TO GET '

.ASCIIZ 'A HIT ON A REFERENCE WHICH SHOULD HAVE BEEN A HIT.'

EM76=EM75

EM77: .ASCII 'CONTROL REGISTER TEST FAILED.'<CRLF>'THE WRONG '

7264	043160	044124	020105	051127	
7265	043166	047117	020107		
7266	043172	051107	052517	020120	.ASCIZ 'GROUP WAS WRITTEN WHILE FORCING SELECTION OF A GROUP.'
7267	043200	040527	020123	051127	
7268	043206	052111	042524	020116	
7269	043214	044127	046111	020105	
7270	043222	047506	041522	047111	
7271	043230	020107	042523	042514	
7272	043236	052103	047511	020116	
7273	043244	043117	040440	043440	.
7274	043252	047522	050125	000056	
7275					
7276	043260	047503	052116	047522	EM117: .ASCII 'CONTROL REGISTER TEST FAILED.'<CRLF>
7277	043266	020114	042522	044507	
7278	043274	052123	051105	052040	
7279	043302	051505	020124	040506	
7280	043310	046111	042105	100056	
7281	043316	047507	020124	020101	.ASCIZ 'GOT A HIT IN THE GROUP TO WHICH MISSES ARE BEING FORCED.'
7282	043324	044510	020124	047111	
7283	043332	052040	042510	043440	
7284	043340	047522	050125	052040	
7285	043346	020117	044127	041511	
7286	043354	020110	044515	051523	
7287	043362	051505	040440	042522	
7288	043370	041040	044505	043516	
7289	043376	043040	051117	042503	
7290	043404	027104	000		
7291					
7292	043407	110	052111	046457	EM120: .ASCII 'HIT/MISS REGISTER PATTERNS TEST FAILED.'
7293	043414	051511	020123	042522	
7294	043422	044507	052123	051105	
7295	043430	050040	052101	042524	
7296	043436	047122	020123	042524	
7297	043444	052123	043040	044501	
7298	043452	042514	027104		
7299	043456	051200	040505	020104	.ASCII <CRLF>'READ WRONG DATA FROM THE HIT/MISS REGISTER'<CRLF>
7300	043464	051127	047117	020107	
7301	043472	040504	040524	043040	
7302	043500	047522	020115	044124	
7303	043506	020105	044510	027524	
7304	043514	044515	051523	051040	
7305	043522	043505	051511	042524	
7306	043530	100122			
7307	043532	044127	046111	020105	.ASCIZ 'WHILE FLOATING A PATTERN OF HITS AND MISSES THROUGH IT.'
7308	043540	046106	040517	044524	
7309	043546	043516	040440	050040	
7310	043554	052101	042524	047122	
7311	043562	047440	020106	044510	
7312	043570	051524	040440	042116	
7313	043576	046440	051511	042523	
7314	043604	020123	044124	047522	
7315	043612	043525	020110	052111	
7316	043620	000056			
7317					
7318	043622	040503	044103	020105	EM121: .ASCII /CACHE CONTROL SIGNAL, THE 'RANDOM' SIGNAL, TEST FAILED./
7319	043630	047503	052116	047522	

7320	043636	020114	044523	047107
7321	043644	046101	020054	044124
7322	043652	020105	051047	047101
7323	043660	047504	023515	051440
7324	043666	043511	040516	026114
7325	043674	052040	051505	020124
7326	043702	040506	046111	042105
7327	043710	056		
7328	043711	200	040506	046111
7329	043716	042105	052040	020117
7330	043724	042507	020124	047507
7331	043732	044124	044040	052111
7332	043740	020123	052101	052040
7333	043746	042510	052040	047527
7334	043754	052040	051505	020124
7335	043762	042101	051104	051505
7336	043770	042523	020123	
7337	043774	044127	041511	020110
7338	044002	042527	042522	051040
7339	044010	043105	051105	047105
7340	044016	042503	027104	000
7341				
7342	044023	115	044501	052116
7343	044030	047105	047101	042503
7344	044036	051040	043505	051511
7345	044044	042524	020122	047503
7346	044052	047125	020124	040520
7347	044060	052124	051105	020116
7348	044066	042524	052123	043040
7349	044074	044501	042514	027104
7350	044102	052200	042510	046440
7351	044110	044501	052116	047105
7352	044116	047101	042503	051040
7353	044124	043505	051511	042524
7354	044132	020122	044527	046114
7355	044140	047040	052117	041440
7356	044146	042514	051101	056
7357				
7358	044153	103	041501	042510
7359	044160	046440	044501	052116
7360	044166	047105	047101	042503
7361	044174	051040	043505	051511
7362	044202	042524	020122	047503
7363	044210	047125	020124	040520
7364	044216	052124	051105	020116
7365	044224	042524	052123	043040
7366	044232	044501	042514	027104
7367	044240	040600	052106	051105
7368	044246	053440	044522	044524
7369	044254	043516	040440	050040
7370	044262	052101	042524	047122
7371	044270	044440	020116	044124
7372	044276	051511	051040	043505
7373	044304	051511	042524	020122
7374	044312	040506	046111	042105
7375	044320	052040	020117	042522

.ASCII <CRLF>'FAILED TO GET BOTH HITS AT THE TWO TEST ADDRESSES '

.ASCIIZ 'WHICH WERE REFERENCED.'

EM122: .ASCII 'MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'

.ASCII <CRLF>'THE MAINTENANCE REGISTER WILL NOT CLEAR.'

EM123: .ASCII 'CACHE MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'

.ASCII <CRLF>'AFTER WRITING A PATTERN IN THIS REGISTER '

.ASCIIZ 'FAILED TO READ THAT PATTERN BACK.'

7376	044326	042101	052040	040510	
7377	044334	020124	040520	052124	
7378	044342	051105	020116	040502	
7379	044350	045503	000056		
7380					
7381	044354	047101	052440	042516	EM124: .ASCII 'AN UNEXPECTED ERROR OCCURRED WHILE RUNNING THE '
7382	044362	050130	041505	042524	
7383	044370	020104	051105	047522	
7384	044376	020122	041517	052503	
7385	044404	051122	042105	053440	
7386	044412	044510	042514	051040	
7387	044420	047125	044516	043516	
7388	044426	052040	042510	040	
7389	044433	115	044501	052116	.ASCII 'MAINTENANCE REGISTER'<CRLF>'COUNT PATTERN '
7390	044440	047105	047101	042503	
7391	044446	051040	043505	051511	
7392	044454	042524	100122	047503	
7393	044462	047125	020124	040520	
7394	044470	052124	051105	020116	.ASCIIZ 'TEST. NOTE MISSES WERE BEING FORCED TO BOTH GROUPS.'
7395	044476	042524	052123	020056	
7396	044504	047516	042524	046440	
7397	044512	051511	042523	020123	
7398	044520	042527	042522	041040	
7399	044526	044505	043516	043040	
7400	044534	051117	042503	020104	
7401	044542	047524	041040	052117	
7402	044550	020110	051107	052517	
7403	044556	051520	000056		
7404					
7405	044562	040515	047111	042524	EM127: .ASCII 'MAINTENANCE REGISTER TEST FAILED.'<CRLF>
7406	044570	040516	041516	020105	
7407	044576	042522	044507	052123	
7408	044604	051105	052040	051505	
7409	044612	020124	040506	046111	
7410	044620	042105	100056		
7411	044624	047516	052040	040522	.ASCII 'NO TRAP OR ABORT OCCURRED WHEN THE PATTERN WAS PUT '
7412	044632	020120	051117	040440	
7413	044640	047502	052122	047440	
7414	044646	041503	051125	042522	
7415	044654	020104	044127	047105	
7416	044662	052040	042510	050040	
7417	044670	052101	042524	047122	
7418	044676	053440	051501	050040	
7419	044704	052125	040		
7420	044707	111	020116	044124	.ASCIIZ 'IN THE MAINTENANCE REGISTER.'
7421	044714	020105	040515	047111	
7422	044722	042524	040516	041516	
7423	044730	020105	042522	044507	
7424	044736	052123	051105	000056	
7425					
7426	044744	051105	047522	020122	EM130: .ASCIIZ 'ERROR REGISTER WILL NOT UNLOCK, OR CLEAR.'
7427	044752	042522	044507	052123	
7428	044760	051105	053440	046111	
7429	044766	020114	047516	020124	
7430	044774	047125	047514	045503	
7431	045002	020054	051117	041440	

7432	045010	042514	051101	000056
7433				
7434	045016	051105	047522	020122
7435	045024	042522	044507	052123
7436	045032	051105	040440	042116
7437	045040	046440	044501	052116
7438	045046	047105	047101	042503
7439	045054	051040	043505	051511
7440	045062	042524	020122	042524
7441	045070	052123	043040	044501
7442	045076	042514	027104	
7443	045102	042600	051122	051117
7444	045110	051040	043505	051511
7445	045116	042524	020122	051511
7446	045124	044440	041516	051117
7447	045132	042522	052103	054514
7448	045140	051440	052105	
7449	045144	043200	051117	052040
7450	045152	042510	042440	051122
7451	045160	051117	052040	040510
7452	045166	020124	040527	020123
7453	045174	047506	041522	042105
7454	045202	052440	044523	043516
7455	045210	052040	042510	046440
7456	045216	044501	052116	047105
7457	045224	047101	042503	051040
7458	045232	043505	051511	042524
7459	045240	027122	000	
7460				
7461	045243			
7462	045243	115	044501	020116
7463	045250	042515	047515	054522
7464	045256	042040	052101	020101
7465	045264	040520	044522	054524
7466	045272	041440	042510	045503
7467	045300	051105	020123	042524
7468	045306	052123	043040	044501
7469	045314	042514	027104	
7470	045320	052600	040516	046102
7471	045326	020105	047524	043040
7472	045334	051117	042503	040440
7473	045342	050040	051101	052111
7474	045350	020131	051105	047522
7475	045356	026122	052440	044523
7476	045364	043516	040	
7477	045367	124	042510	046440
7478	045374	044501	052116	047105
7479	045402	047101	042503	051040
7480	045410	043505	051511	042524
7481	045416	026122	200	
7482	045421	101	020124	044124
7483	045426	020105	040515	047111
7484	045434	046440	046505	051117
7485	045442	020131	053105	047105
7486	045450	053440	051117	026104
7487	045456	046040	053517	041040

EM131: .ASCII 'ERROR REGISTER AND MAINTENANCE REGISTER TEST FAILED.'

.ASCII <CRLF>'ERROR REGISTER IS INCORRECTLY SET'

.ASCIIZ <CRLF>'FOR THE ERROR THAT WAS FORCED USING THE MAINTENANCE REGISTER.'

EM140: .ASCII 'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'

.ASCII <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '

.ASCII 'THE MAINTENANCE REGISTER,'<CRLF>

.ASCII 'AT THE MAIN MEMORY EVEN WORD, LOW BYTE, PARITY '

7488	045464	052131	026105	050040	
7489	045472	051101	052111	020131	
7490	045500	044103	041505	042513	.ASCII 'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
7491	045506	026122	020200	042522	
7492	045514	042101	047111	020107	
7493	045522	020101	040504	040524	
7494	045530	050040	052101	042524	
7495	045536	047122	053440	044510	
7496	045544	044103	040		
7497	045547	123	047510	046125	.ASCIZ 'SHOULD HAVE CAUSED AN ERROR.'
7498	045554	020104	040510	042526	
7499	045562	041440	052501	042523	
7500	045570	020104	047101	042440	
7501	045576	051122	051117	000056	
7502					
7503	045604				EM141:
7504	045604	040515	047111	046440	.ASCII 'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
7505	045612	046505	051117	020131	
7506	045620	040504	040524	050040	
7507	045626	051101	052111	020131	
7508	045634	044103	041505	042513	
7509	045642	051522	052040	051505	
7510	045650	020124	040506	046111	
7511	045656	042105	056		
7512	045661	200	047125	041101	.ASCII <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
7513	045666	042514	052040	020117	
7514	045674	047506	041522	020105	
7515	045702	020101	040520	044522	
7516	045710	054524	042440	051122	
7517	045716	051117	020054	051525	
7518	045724	047111	020107		
7519	045730	044124	020105	040515	.ASCII 'THE MAINTENANCE REGISTER,'<CRLF>
7520	045736	047111	042524	040516	
7521	045744	041516	020105	042522	
7522	045752	044507	052123	051105	
7523	045760	100054			
7524	045762	052101	052040	042510	.ASCII 'AT THE MAIN MEMORY ODD WORD, LOW BYTE, PARITY '
7525	045770	046440	044501	020116	
7526	045776	042515	047515	054522	
7527	046004	047440	042104	053440	
7528	046012	051117	026104	046040	
7529	046020	053517	041040	052131	
7530	046026	026105	050040	051101	
7531	046034	052111	020131		
7532	046040	044103	041505	042513	.ASCII 'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
7533	046046	026122	020200	042522	
7534	046054	042101	047111	020107	
7535	046062	020101	040504	040524	
7536	046070	050040	052101	042524	
7537	046076	047122	053440	044510	
7538	046104	044103	040		
7539	046107	123	047510	046125	.ASCIZ 'SHOULD HAVE CAUSED AN ERROR.'
7540	046114	020104	040510	042526	
7541	046122	041440	052501	042523	
7542	046130	020104	047101	042440	
7543	046136	051122	051117	000056	

```

7544
7545 046144
7546 046144 040515 047111 046440
7547 046152 046505 051117 020131
7548 046160 040504 040524 050040
7549 046166 051101 052111 020131
7550 046174 044103 041505 042513
7551 046202 051522 052040 051505
7552 046210 020124 040506 046111
7553 046216 042105 056
7554 046221 200 047125 041101
7555 046226 042514 052040 020117
7556 046234 047506 041522 020105
7557 046242 020101 040520 044522
7558 046250 054524 042440 051122
7559 046256 051117 020054 051525
7560 046264 047111 020107
7561 046270 044124 020105 040515
7562 046276 047111 042524 040516
7563 046304 041516 020105 042522
7564 046312 044507 052123 051105
7565 046320 100054
7566 046322 052101 052040 042510
7567 046330 046440 044501 020116
7568 046336 042515 047515 054522
7569 046344 042440 042526 020116
7570 046352 047527 042122 020054
7571 046360 044510 044107 041040
7572 046366 052131 026105 050040
7573 046374 051101 052111 020131
7574 046402 044103 041505 042513
7575 046410 026122 020200 042522
7576 046416 042101 047111 020107
7577 046424 020101 040504 040524
7578 046432 050040 052101 042524
7579 046440 047122 053440 044510
7580 046446 044103 040
7581 046451 123 047510 046125
7582 046456 020104 040510 042526
7583 046464 041440 052501 042523
7584 046472 020104 047101 042440
7585 046500 051122 051117 000056
7586
7587 046506
7588 046506 040515 047111 046440
7589 046514 046505 051117 020131
7590 046522 040504 040524 050040
7591 046530 051101 052111 020131
7592 046536 044103 041505 042513
7593 046544 051522 052040 051505
7594 046552 020124 040506 046111
7595 046560 042105 056
7596 046563 200 047125 041101
7597 046570 042514 052040 020117
7598 046576 047506 041522 020105
7599 046604 020101 040520 044522

```

EM142:

```

.ASCII 'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
.ASCII <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
.ASCII 'THE MAINTENANCE REGISTER, '<CRLF>
.ASCII 'AT THE MAIN MEMORY EVEN WORD, HIGH BYTE, PARITY '
.ASCII 'CHECKER, '<CRLF>' READING A DATA PATTERN WHICH '
.ASCIIZ 'SHOULD HAVE CAUSED AN ERROR.'

```

EM143:

```

.ASCII 'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
.ASCII <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '

```

7600	046612	054524	042440	051122	
7601	046620	051117	020054	051525	
7602	046626	047111	020107		
7603	046632	044124	020105	040515	.ASCII 'THE MAINTENANCE REGISTER,<CRLF>
7604	046640	047111	042524	040516	
7605	046646	041516	020105	042522	
7606	046654	044507	052123	051105	
7607	046662	100054			
7608	046664	052101	052040	042510	.ASCII 'AT THE MAIN MEMORY ODD WORD, HIGH BYTE, PARITY '
7609	046672	046440	044501	020116	
7610	046700	042515	047515	054522	
7611	046706	047440	042104	053440	
7612	046714	051117	026104	044040	
7613	046722	043511	020110	054502	
7614	046730	042524	020054	040520	
7615	046736	044522	054524	040	
7616	046743	103	042510	045503	.ASCII 'CHECKER,<CRLF>' READING A DATA PATTERN WHICH '
7617	046750	051105	100054	051040	
7618	046756	040505	044504	043516	
7619	046764	040440	042040	052101	
7620	046772	020101	040520	052124	
7621	047000	051105	020116	044127	
7622	047006	041511	020110		
7623	047012	044123	052517	042114	.ASCIIZ 'SHOULD HAVE CAUSED AN ERROR.'
7624	047020	044040	053101	020105	
7625	047026	040503	051525	042105	
7626	047034	040440	020116	051105	
7627	047042	047522	027122	000	
7628					
7629	047047	040	052040	051505	DH140: .ASCIIZ ' TEST.<TAB>'CALL AT PC.<TAB>'DATA.<TAB>'ADDRESS.'
7630	047054	027124	041411	046101	
7631	047062	020114	052101	050040	
7632	047070	027103	042011	052101	
7633	047076	027101	040411	042104	
7634	047104	042522	051523	000056	
7635					
7636		047047			DH141=DH140
7637					
7638		047047			DH142=DH140
7639					
7640		047047			DH143=DH140
7641					
7642	047112	004	003	000	DF140: .BYTE 4,3,0,2
7643	047115	002			
7644					
7645		047112			DF141-DF140
7646					
7647		047112			DF142=DF140
7648					
7649		047112			DF143-DF140
7650					
7651					.EVEN
7652	047116	001224	001116	001230	DT140: .WORD \$TMP0,\$ERRPC,\$TMP2,\$TMP3,0
7653	047124	001232	000000		
7654					
7655		047116			DT141=DT140



```
7656
7657      047116      DT142=DT140
7658
7659      047116      DT143=DT140
7660
7661
7662 047130 051105 047522 020122 EM132: .ASCII 'ERROR REGISTER TEST WAS UNABLE TO CAUSE A TIME OUT.'
7663 047136 042522 044507 052123
7664 047144 051105 052040 051505
7665 047152 020124 040527 020123
7666 047160 047125 041101 042514
7667 047166 052040 020117 040503
7668 047174 051525 020105 020101
7669 047202 044524 042515 047440
7670 047210 052125      054
7671 047213      200 052101 040440      .ASCIIZ <CRLF>'AT AN ADDRESS WHICH SHOULD HAVE TIMED OUT.'
7672 047220 020116 042101 051104
7673 047226 051505 020123 044127
7674 047234 041511 020110 044123
7675 047242 052517 042114 044040
7676 047250 053101 020105 044524
7677 047256 042515 020104 052517
7678 047264 027124      000
7679
7680 047267      105 051122 051117 EM133: .ASCII 'ERROR REGISTER TEST FAILED.'
7681 047274 051040 043505 051511
7682 047302 042524 020122 042524
7683 047310 052123 043040 044501
7684 047316 042514 027104
7685 047322 040600 052106 051105      .ASCII <CRLF>'AFTER CAUSING A TIME OUT THE ERROR REGISTER SHOULD '
7686 047330 041440 052501 044523
7687 047336 043516 040440 052040
7688 047344 046511 020105 052517
7689 047352 020124 044124 020105
7690 047360 051105 047522 020122
7691 047366 042522 044507 052123
7692 047374 051105 051440 047510
7693 047402 046125 020104
7694 047406 040510 042526 041040      .ASCIIZ 'HAVE BEEN SET TO : 000000.'
7695 047414 042505 020116 042523
7696 047422 020124 047524 035040
7697 047430 030040 030060 030060
7698 047436 027060      000
7699
7700 047441      103 047117 051124 EM134: .ASCII 'CONTROL REGISTER, DISABLE TRAPS, TEST FAILED.'
7701 047446 046117 051040 043505
7702 047454 051511 042524 026122
7703 047462 042040 051511 041101
7704 047470 042514 052040 040522
7705 047476 051520 020054 042524
7706 047504 052123 043040 044501
7707 047512 042514 027104
7708 047516 040600 052040 040522      .ASCIIZ <CRLF>'A TRAP OCCURRED WITH BIT 0 SET IN THE CONTROL REGISTER.'
7709 047524 020120 041517 052503
7710 047532 051122 042105 053440
7711 047540 052111 020110 044502
```



7768	050202	052101	050040	027103	
7769	050210	011			
7770	050211	105	051122	051117	.ASCIZ 'ERROR REG.'
7771	050216	051040	043505	000056	
7772					
7773	050224	020040	042524	052123	DH15: .ASCIZ ' TEST.<TAB>'CALL AT PC.'
7774	050232	004456	040503	046114	
7775	050240	040440	020124	041520	
7776	050246	000056			
7777					
7778	050250	020040	042524	052123	DH55: .ASCIZ ' TEST.<TAB>'TRAP AT PC.<TAB>'CALL AT PC.<TAB>'REG ADDRESS.'
7779	050256	004456	051124	050101	
7780	050264	040440	020124	041520	
7781	050272	004456	040503	046114	
7782	050300	040440	020124	041520	
7783	050306	004456	042522	020107	
7784	050314	042101	051104	051505	
7785	050322	027123	000		
7786					
7787		050250			DH56-DH55
7788					
7789		050250			DH57=DH55
7790					
7791		050250			DH60=DH55
7792					
7793		050250			DH61=DH55
7794					
7795		050250			DH62-DH55
7796					
7797	050325	040	052040	051505	DH63: .ASCII ' TEST.<TAB>'CALL AT PC.<TAB>'CONTROL.'
7798	050332	027124	041411	046101	
7799	050340	020114	052101	050040	
7800	050346	027103	041411	047117	
7801	050354	051124	046117	056	
7802	050361	115	044501	052116	.ASCIZ 'MAINT.<TAB>'(DATA READ FROM EACH REGISTER)'
7803	050366	004456	042050	052101	
7804	050374	020101	042522	042101	
7805	050402	043040	047522	020115	
7806	050410	040505	044103	051040	
7807	050416	043505	051511	042524	
7808	050424	024522	000		
7809					
7810	050427	040	052040	051505	DH64: .ASCIZ ' TEST.<TAB>'CALL AT PC.<TAB>'CONTROL REGISTER DATA.'
7811	050434	027124	041411	046101	
7812	050442	020114	052101	050040	
7813	050450	027103	041411	047117	
7814	050456	051124	046117	051040	
7815	050464	043505	051511	042524	
7816	050472	020122	040504	040524	
7817	050500	000056			
7818					
7819	050502	020040	042524	052123	DH65: .ASCII ' TEST.<TAB>'CALL AT PC.<TAB>'LOW ORD.<TAB>'HIGH ORD.'
7820	050510	004456	040503	046114	
7821	050516	040440	020124	041520	
7822	050524	004456	047514	020127	
7823	050532	051117	027104	044011	

7824	050540	043511	020110	051117	
7825	050546	027104			
7826	050550	024011	040504	040524	.ASCIZ <TAB>'(DATA READ FROM ADR. REG.)'
7827	050556	051040	040505	020104	
7828	050564	051106	046517	040440	
7829	050572	051104	020056	042522	
7830	050600	027107	000051		
7831					
7832	050604	020040	042524	052123	DH66: .ASCII ' TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>'READ.'
7833	050612	004456	040503	046114	
7834	050620	040440	020124	041520	
7835	050626	004456	051127	052117	
7836	050634	027105	051011	040505	
7837	050642	027104			
7838	050644	042411	050130	041505	.ASCIZ <TAB>'EXPECTED.'
7839	050652	042524	027104	000	
7840					
7841	050657	040	052040	051505	DH67: .ASCII ' TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN READ FROM THE '
7842	050664	027124	041411	046101	
7843	050672	020114	052101	050040	
7844	050700	027103	050011	052101	
7845	050706	042524	047122	051040	
7846	050714	040505	020104	051106	
7847	050722	046517	052040	042510	
7848	050730	040			
7849	050731	110	052111	046457	.ASCIZ 'HIT/MISS REGISTER.'
7850	050736	051511	020123	042522	
7851	050744	044507	052123	051105	
7852	050752	000056			
7853					
7854		050657			DH70=DH67
7855					
7856		050657			DH71=DH67
7857					
7858		050657			DH72=DH67
7859					
7860		050657			DH73=DH67
7861					
7862		050657			DH74=DH67
7863					
7864	050754	020040	042524	052123	DH75: .ASCII ' TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>
7865	050762	004456	040503	046114	
7866	050770	040440	020124	041520	
7867	050776	004456	043440	047522	
7868	051004	050125	004456		
7869	051010	042101	051104	051505	.ASCIZ 'ADDRESS.'<TAB>'PATTERN IN CONTROL REG.'
7870	051016	027123	050011	052101	
7871	051024	042524	047122	044440	
7872	051032	020116	047503	052116	
7873	051040	047522	020114	042522	
7874	051046	027107	000		
7875					
7876		050754			DH76=DH75
7877					
7878	051051	040	052040	051505	DH77: .ASCIZ ' TEST.'<TAB>'CALL AT PC.'
7879	051056	027124	041411	046101	

7880	051064	020114	052101	050040	
7881	051072	027103	000		
7882					
7883					
7884		050754			DH117=DH75
7885					
7886	051075	040	052040	051505	DH120: .ASCIZ ' TEST.<TAB>'CALL AT PC.<TAB>'PATTERN IN CONTROL REG.'
7887	051102	027124	041411	046101	
7888	051110	020114	052101	050040	
7889	051116	027103	050011	052101	
7890	051124	042524	047122	044440	
7891	051132	020116	047503	052116	
7892	051140	047522	020114	042522	
7893	051146	027107	000		
7894					
7895	051151	040	052040	051505	DH121: .ASCIZ ' TEST.<TAB>'CALL AT PC.<TAB>'TEST ADDRESS.'
7896	051156	027124	041411	046101	
7897	051164	020114	052101	050040	
7898	051172	027103	052011	051505	
7899	051200	020124	042101	051104	
7900	051206	051505	027123	000	
7901					
7902	051213	040	052040	051505	DH122: .ASCII ' TEST.<TAB>'CALL AT PC.<TAB>'WROTE.<TAB>
7903	051220	027124	041411	046101	
7904	051226	020114	052101	050040	
7905	051234	027103	053411	047522	
7906	051242	042524	004456		
7907	051246	044124	047105	041440	.ASCIZ 'THEN CLEARED AND READ.'
7908	051254	042514	051101	042105	
7909	051262	040440	042116	051040	
7910	051270	040505	027104	000	
7911					
7912	051275	040	042524	052123	DH123: .ASCIZ ' TEST.<TAB>'CALL AT PC.<TAB>'WROTE.<TAB>'READ.'
7913	051302	004456	040503	046114	
7914	051310	040440	020124	041520	
7915	051316	004456	051127	052117	
7916	051324	027105	051011	040505	
7917	051332	027104	000		
7918					
7919		050131			DH124=DH14
7920					
7921	051335	040	052040	051505	DH125: .ASCIZ ' TEST.<TAB>'CALL AT PC.<TAB>'ADDRESS.'
7922	051342	027124	041411	046101	
7923	051350	020114	052101	050040	
7924	051356	027103	040411	042104	
7925	051364	042522	051523	000056	
7926					
7927	051372	020040	042524	052123	DH126: .ASCII ' TEST.<TAB>'CALL AT PC.<TAB>'TRAP AT PC.'
7928	051400	004456	040503	046114	
7929	051406	040440	020124	041520	
7930	051414	004456	051124	050101	
7931	051422	040440	020124	041520	
7932	051430	056			
7933	051431	011	051105	047522	.ASCIZ <TAB>'ERROR REG.'
7934	051436	020122	042522	027107	
7935	051444	000			

```
7936
7937 051445 040 052040 051505 DH127: .ASCIZ ' TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN USED.'
7938 051452 027124 041411 046101
7939 051460 020114 052101 050040
7940 051466 027103 050011 052101
7941 051474 042524 047122 052440
7942 051502 042523 027104 000
7943
7944 051507 040 052040 051505 DH130: .ASCII ' TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADR REG.'
7945 051514 027124 041411 046101
7946 051522 020114 052101 050040
7947 051530 027103 042411 051122
7948 051536 051117 040440 051104
7949 051544 051040 043505 056
7950 051551 011 051105 047522 .ASCIZ <TAB>'ERROR REG.'
7951 051556 020122 042522 027107
7952 051564 000
7953
7954 051565 040 052040 051505 DH131: .ASCII ' TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
7955 051572 027124 041411 046101
7956 051600 020114 052101 050040
7957 051606 027103 052011 040522
7958 051614 020120 052101 050040
7959 051622 027103 011
7960 051625 105 051122 051117 .ASCIZ 'ERROR ADR REG.'
7961 051632 040440 051104 051040
7962 051640 043505 000056
7963
7964 051335 DH132=DH125
7965
7966 051372 DH133=DH126
7967
7968 051644 020040 042524 052123 DH134: .ASCII ' TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
7969 051652 004456 040503 046114
7970 051660 040440 020124 041520
7971 051666 004456 051124 050101
7972 051674 040440 020124 041520
7973 051702 004456
7974 051704 047503 052116 047522 .ASCIZ 'CONTROL REG.'
7975 051712 020114 042522 027107
7976 051720 000
7977
7978 051051 DH135=DH77
7979
7980 051721 040 052040 051505 DH150: .ASCIZ ' TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'CPU ERROR REGISTER.'
7981 051726 027124 052011 040522
7982 051734 020120 052101 050040
7983 051742 027103 041411 046101
7984 051750 020114 052101 050040
7985 051756 027103 041411 052520
7986 051764 042440 051122 051117
7987 051772 051040 043505 051511
7988 052000 042524 027122 000
7989 ;THESE ARE DATA FORMAT DESIGNATORS FOR THE DATA TABLE:
7990
7991 052005 004 004 003 DF1: .BYTE 4,4,3,3
```

7992	052010	003						
7993								
7994	052011	004	003	007	DF14:	.BYTE	4,3,7,3,0	
7995	052014	003	000					
7996								
7997	052016	004	003		DF15:	.BYTE	4,3	
7998								
7999	052020	004	003	003	DF55:	.BYTE	4,3,3,2	
8000	052023	002						
8001								
8002		052020			DF56=DF55			
8003								
8004		052020			DF57=DF55			
8005								
8006		052020			DF60=DF55			
8007								
8008		052020			DF61=DF55			
8009								
8010		052020			DF62=DF55			
8011								
8012	052024	004	003	000	DF63:	.BYTE	4,3,0,0,0	
8013	052027	000	000					
8014								
8015		052024			DF64=DF63			
8016								
8017		052024			DF65=DF63			
8018								
8019		052024			DF66=DF63			
8020								
8021		052024			DF67=DF63			
8022								
8023		052024			DF70=DF63			
8024								
8025		052024			DF71=DF63			
8026								
8027		052024			DF72=DF63			
8028								
8029		052024			DF73=DF63			
8030								
8031		052024			DF74=DF63			
8032								
8033	052031	004	003	004	DF75:	.BYTE	4,3,4,2,0	
8034	052034	002	000					
8035								
8036		052031			DF76=DF75			
8037								
8038	052036	004	003	005	DF77:	.BYTE	4,3,5,2,5,0,5,2,5,0	
8039	052041	002	005	000				
8040	052044	005	002	005				
8041	052047	000						
8042								
8043								
8044		052031			DF117=DF75			
8045								
8046	052050	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	
8047	052053	005	000	005				

8048	052056	000	005	000			
8049	052061	005	000	005			
8050	052064	000	005	000			
8051	052067	005	000	005			
8052	052072	000	005	000			
8053	052075	005	000	005			
8054	052100	000	005	000			
8055							
8056	052103	004	003	002	DF121:	.BYTE	4,3,2,2
8057	052106	002					
8058							
8059	052107	004	003	000	DF122:	.BYTE	4,3,0,0
8060	052112	000					
8061							
8062	052107				DF123=DF122		
8063							
8064	052113	004	003	007	DF124:	.BYTE	4,3,7,3,0,5,0,
8065	052116	003	000	005			
8066	052121	000	000				
8067							
8068	052123	004	003	002	DF125:	.BYTE	4,3,2,0
8069	052126	000					
8070							
8071	052127	004	003	003	DF126:	.BYTE	4,3,3,0,5,2,5,2
8072	052132	000	005	002			
8073	052135	005	002				
8074							
8075	052137	004	003	000	DF127:	.BYTE	4,3,0
8076							
8077	052123				DF130=DF125		
8078							
8079	052142	004	003	003	DF131:	.BYTE	4,3,3,2,5,0,5,0,5,0
8080	052145	002	005	000			
8081	052150	005	000	005			
8082	052153	000					
8083							
8084	052123				DF132=DF125		
8085							
8086	052127				DF133=DF126		
8087							
8088	052154	004	003	003	DF134:	.BYTE	4,3,3,0,5,2,0
8089	052157	000	005	002			
8090	052162	000					
8091							
8092	052163	004	003	005	DF135:	.BYTE	4,3,5,0,5,0,5,2,5,2
8093	052166	000	005	000			
8094	052171	005	002	005			
8095	052174	002					
8096							
8097	052175	004	003	003	DF150:	.BYTE	4,3,3,0
8098	052200	000					
8099							
8100	052202				.EVEN		
8101							
8102					; THESE ARE DATA TABLES:		
8103							



8104	052202	001224	001226	001230	DT1:	.WORD	\$TMP0,\$TMP1,\$TMP2,\$ERRPC,0
8105	052210	001116	000000				
8106							
8107	052214	001224	001116	001226	DT14:	.WORD	\$TMP0,\$ERRPC,\$TMP1,\$TMP3,\$TMP4,0
8108	052222	001232	001234	000000			
8109							
8110	052230	001224	001226	000000	DT15:	.WORD	\$TMP0,\$TMP1,0
8111							
8112							
8113	052236	001224	001226	001116	DT55:	.WORD	\$TMP0,\$TMP1,\$ERRPC,\$TMP3,0
8114	052244	001232	000000				
8115							
8116		052236			DT56=	DT55	
8117							
8118		052236			DT57=	DT55	
8119							
8120		052236			DT60=	DT55	
8121							
8122		052236			DT61=	DT55	
8123							
8124		052236			DT62=	DT55	
8125							
8126	052250	001224	001116	001230	DT63:	.WORD	\$TMP0,\$ERRPC,\$TMP2,\$TMP3,0
8127	052256	001232	000000				
8128							
8129	052262	001224	001116	001230	DT64:	.WORD	\$TMP0,\$ERRPC,\$TMP2,0
8130	052270	000000					
8131							
8132	052272	001224	001116	001230	DT65:	.WORD	\$TMP0,\$ERRPC,\$TMP2,\$TMP3,0
8133	052300	001232	000000				
8134							
8135	052304	001224	001116	001230	DT66:	.WORD	\$TMP0,\$ERRPC,\$TMP2,\$TMP3,\$TMP4,0
8136	052312	001232	001234	000000			
8137							
8138		052262			DT67=	DT64	
8139							
8140		052262			DT70=	DT64	
8141							
8142		052262			DT71=	DT64	
8143							
8144		052262			DT72=	DT64	
8145							
8146		052262			DT73=	DT64	
8147							
8148		052262			DT74=	DT64	
8149							
8150	052320	001224	001116	001230	DT75:	.WORD	\$TMP0,\$ERRPC,\$TMP2,\$TMP10,\$TMP3,0
8151	052326	001244	001232	000000			
8152							
8153	052334	001224	001116	001230	DT76:	.WORD	\$TMP0,\$ERRPC,\$TMP2,\$TMP12,\$TMP3,0
8154	052342	001250	001232	000000			
8155							
8156	052350	001224	001116	035	DT77:	.WORD	\$TMP0,\$ERRPC,MTA77,\$TMP10,MTB77,\$TMP2,MTC77
8157	052356	001244	035124	001230			
8158	052364	035166					
8159	052366	001250	035223	001232		.WORD	\$TMP12,MTD77,\$TMP5,0

C  
A  
R  
D  
I  
M  
A  
G  
E  
S  
A  
R  
E  
N  
O  
T  
T  
O  
B  
E  
A  
R  
N  
G  
T  
O  
T  
H  
I  
S  
D  
O  
C  
U  
M  
E  
N  
T  
.

```

8160 052374 000000
8161
8162 052334 DT117=DT76
8163
8164 052376 001224 001116 001230 DT120: .WORD $TMP0,$ERRPC,$TMP2,MTA120,KCR0,MTG120,KCE0
8165 052404 035373 007670 035613
8166 052412 007704
8167 052414 035423 007672 035613 .WORD MTB120,KCR1,MTG120,KCE1
8168 052422 007706
8169 052424 035453 007674 035613 .WORD MTC120,KCR2,MTG120,KCE2
8170 052432 007710
8171 052434 035503 007676 035613 .WORD MTD120,KCR3,MTG120,KCE3
8172 052442 007712
8173 052444 035533 007700 035613 .WORD MTE120,KCR4,MTG120,KCE4
8174 052452 007714
8175 052454 035563 007702 035613 .WORD MTF120,KCR5,MTG120,KCE5,0
8176 052462 007716 000000
8177
8178 052466 001224 001116 001230 DT121: .WORD $TMP0,$ERRPC,$TMP2,$TMP4,0
8179 052474 001234 000000
8180
8181 052500 001224 001116 001230 DT122: .WORD $TMP0,$ERRPC,$TMP2,$TMP3,0
8182 052506 001232 000000
8183
8184 052500 DT123=DT122
8185
8186 052512 001224 001116 001226 DT124: .WORD $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,MTA124,$TMP6,0
8187 052520 001232 001234 035654
8188 052526 001240 000000
8189
8190 052532 001224 001116 001230 DT125: .WORD $TMP0,$ERRPC,$TMP2,0
8191 052540 000000
8192
8193 052542 001224 001116 001230 DT126: .WORD $TMP0,$ERRPC,$TMP2,$TMP7,MTA126,$TMP5,MTB126,$TMP3,0
8194 052550 001242 035746 001236
8195 052556 035774 001232 000000
8196
8197 052532 DT127=DT125
8198
8199 052564 001224 001116 001230 DT130: .WORD $TMP0,$ERRPC,$TMP2,$TMP4,0
8200 052572 001234 000000
8201
8202 052576 001224 001116 001230 DT131: .WORD $TMP0,$ERRPC,$TMP2,$TMP3,MTA131,$TMP5
8203 052604 001232 036026 001236
8204 052612 036110 001240 036143 .WORD MTB131,$TMP6,MTC131,$TMP7,0
8205 052620 001242 000000
8206
8207 052532 DT132=DT125
8208
8209 052542 DT133=DT126
8210
8211 052624 001224 001116 001230 DT134: .WORD $TMP0,$ERRPC,$TMP2,$TMP3,MTA134,$TMP4,$TMP6,0
8212 052632 001232 036171 001234
8213 052640 001240 000000
8214
8215 052644 001224 001116 036225 DT135: .WORD $TMP0,$ERRPC,MTA135,$TMP2,MTB135,$TMP3

```

8216	052652	001230	036255	001232		
8217	052660	036277	001234	036333	.WORD	MTC135,\$TMP4,MTD135,\$TMP6,0
8218	052666	001240	000000			
8219						
8220	052672	001224	001226	001230	DT150: .WORD	\$TMP0,\$TMP1,\$TMP2,\$TMP3,0
8221	052700	001232	000000			
8222						
8223	052704	000000	000000	000000	BOTTOM: .WORD	0,0,0
8224		060712				.=.+6000
8225	060712				BOTPRG:	
8226		000001				.END

ABORTT	032062	5892	6109#											
ADRNG	034365	6155	6583#											
BACKAD	032200	6125*	6134	6135#										
BIT0 =	000001	138#	973	974	981	1334	1425	1523	4472	4546	4611	5217	5338	6025
BIT00 =	000001	128#	138											
BIT01 =	000002	127#	137											
BIT02 =	000004	126#	136											
BIT03 =	000010	125#	135											
BIT04 =	000020	124#	134											
BIT05 =	000040	123#	133											
BIT06 =	000100	122#	132											
BIT07 =	000200	121#	131	1032										
BIT08 =	000400	120#	130	5501										
BIT09 =	001000	119#	129	5512	5578									
BIT1 =	000002	137#	1430	1528										
BIT10 =	002000	118#	5561											
BIT11 =	004000	117#	976	977	979	5519								
BIT12 =	010000	116#												
BIT13 =	020000	115#	5568											
BIT14 =	040000	114#	968	969	971									
BIT15 =	100000	113#	982	983	985	987	988	990	6028					
BIT2 =	000004	136#	1435	1533										
BIT3 =	000010	135#												
BIT4 =	000020	134#												
BIT5 =	000040	133#												
BIT6 =	000100	132#	1090	6251										
BIT7 =	000200	131#	6230											
BIT8 =	000400	130#	1003											
BIT9 =	001000	129#	1330	5503										
BOTPRG	060712	8225#												
BOTTOM	052704	1080	6236	8223#										
BPTVEC	000014	145#												
CACHE	000114	152#	1093*	1110*	1202*	1567*	1600*	1751*	1962*	2073*	2141*	2155*	2182*	2258*
		2343*	2427*	2514*	2595*	2690*	2778*	2877*	2976*	3075*	3174*	3273*	3372*	3471*
		3574*	3698*	3829*	3941*	4052*	4163*	4274*	4359*	4473*	4547*	4612*	4672*	4696*
		4810*	4836*	4952*	4977*	5093*	5119*	5210*	5221*	5243*	5331*	5342*	5364*	6089*
		6095*	6123*											
		5456	6235#											
CHAINQ	032424	562#												
CISP	001313													
CLEAN	032112	5891	6121#											
CNRNG	034575	6169	6610#											
CONCMS	033340	6234	6463#											
CONFLG	032326	1165*	1227*	1234*	1354*	1936*	6165	6191#						
CONFL2	032342	1446*	1542*	1605*	1628*	1698*	1721*	1909*	1934	6197#				
CONTRL-	177746	162#	965	995*	996	1163	1208*	1210	1316	1378*	1393*	1406*	1437*	1475*
		1491*	1504*	1534*	1583*	1596*	1607*	1620*	1630*	1676*	1689*	1700*	1713*	1723*
		1766*	1768*	1775*	1913*	1976*	1994*	2010*	2081*	2154*	2186*	2261*	2346*	2430*
		2517*	2598*	2691*	2781*	2880*	2979*	3078*	3177*	3276*	3375*	3474*	3607*	3723*
		3830*	3942*	4053*	4164*	4275*	4471*	4472*	4499	4531*	4546*	4564	4596*	4611*
		4629	4669*	4807*	4947*	5088*	5212*	5333*	5503*	6130*				
CPSPUR	031736	1092	3575	4409	6081#	6124								
CPUERR	177766	175#	1135	1183*	1186*	4407	5586*	6083	6132*					
CR	- 000015	50#	5699	5709										
CRLF	000200	51#	934	5672	5709	6463	6465	6472	6482	6485	6492	6502	6512	6515
		6525	6531	6534	6539	6546	6551	6558	6568	6571	6578	6583	6597	6610
		6622	6636	6649	6659	6672	6677	6685	6691	6697	6702	6708	6714	6727

		6739	6744	6750	6760	6766	6771	6777	6788	6798	6804	6806	6808	6813
		6829	6843	6862	6881	6897	6913	6930	6937	6956	6964	6983	6991	6997
		7015	7023	7060	7068	7086	7095	7119	7128	7161	7180	7189	7214	7223
		7239	7259	7276	7299	7328	7350	7367	7389	7405	7443	7449	7470	7477
		7490	7512	7519	7532	7554	7561	7574	7596	7603	7616	7671	7685	7708
		7726	7741											
DF1	052005	588	7991#											
DF117	= 052031	826	8044#											
DF120	052050	829	8046#											
DF121	052103	832	8056#											
DF122	052107	835	8059#	8062										
DF123	= 052107	838	8062#											
DF124	052113	841	8064#											
DF125	052123	8068#	8077	8084										
DF126	052127	8071#	8086											
DF127	052137	850	8075#											
DF130	= 052123	853	8077#											
DF131	052142	857	8079#											
DF132	= 052123	860	8084#											
DF133	= 052127	863	8086#											
DF134	052154	866	8088#											
DF135	052163	869	8092#											
DF14	052011	621	7994#											
DF140	047112	878	7642#	7645	7647	7649								
DF141	= 047112	881	7645#											
DF142	= 047112	884	7647#											
DF143	= 047112	887	7649#											
DF15	052016	624	7997#											
DF150	052175	902	8097#											
DF55	052020	722	7999#	8002	8004	8006	8008	8010						
DF56	= 052020	725	8002#											
DF57	= 052020	728	8004#											
DF60	- 052020	731	8006#											
DF61	- 052020	734	8008#											
DF62	= 052020	737	8010#											
DF63	052024	740	8012#	8015	8017	8019	8021	8023	8025	8027	8029	8031		
DF64	- 052024	743	8015#											
DF65	= 052024	746	8017#											
DF66	= 052024	749	8019#											
DF67	= 052024	752	8021#											
DF70	= 052024	755	8023#											
DF71	= 052024	758	8025#											
DF72	= 052024	761	8027#											
DF73	= 052024	764	8029#											
DF74	= 052024	767	8031#											
DF75	052031	771	8033#	8036	8044									
DF76	= 052031	774	8036#											
DF77	052036	777	8038#											
DH1	050056	588	7753#											
DH117	- 050754	826	7884#											
DH120	051075	829	7886#											
DH121	051151	832	7895#											
DH122	051213	835	7902#											
DH123	051275	838	7912#											
DH124	= 050131	841	7919#											
DH125	051335	7921#	7964											

CCCCCCCCCC

DH126	051372	7927#	7966						
DH127	051445	850	7937#						
DH130	051507	853	7944#						
DH131	051565	857	7954#						
DH132	= 051335	860	7964#						
DH133	= 051372	863	7966#						
DH134	051644	866	7968#						
DH135	= 051051	869	7978#						
DH14	050131	621	7761#	7919					
DH140	047047	878	7629#	7636	7638	7640			
DH141	= 047047	881	7636#						
DH142	= 047047	884	7638#						
DH143	= 047047	887	7640#						
DH15	050224	624	7773#						
DH150	051721	902	7980#						
DH55	050250	722	7778#	7787	7789	7791	7793	7795	
DH56	= 050250	725	7787#						
DH57	= 050250	728	7789#						
DH60	= 050250	731	7791#						
DH61	= 050250	734	7793#						
DH62	= 050250	737	7795#						
DH63	050325	740	7797#						
DH64	050427	743	7810#						
DH65	050502	746	7819#						
DH66	050604	749	7832#						
DH67	050657	752	7841#	7854	7856	7858	7860	7862	
DH70	= 050657	755	7854#						
DH71	= 050657	758	7856#						
DH72	= 050657	761	7858#						
DH73	= 050657	764	7860#						
DH74	= 050657	767	7862#						
DH75	050754	771	7864#	7876	7884				
DH76	= 050754	774	7876#						
DH77	051051	777	7878#	7978					
DISPLA=	177570	45#	5533*	5557*					
DT1	052202	588	8104#						
DT117	= 052334	826	8162#						
DT120	052376	829	8164#						
DT121	052466	832	8178#						
DT122	052500	835	8181#	8184					
DT123	= 052500	838	8184#						
DT124	052512	841	8186#						
DT125	052532	8190#	8197	8207					
DT126	052542	8193#	8209						
DT127	= 052532	850	8197#						
DT130	052564	853	8199#						
DT131	052576	857	8202#						
DT132	= 052532	860	8207#						
DT133	= 052542	863	8209#						
DT134	052624	866	8211#						
DT135	052644	869	8215#						
DT14	052214	621	8107#						
DT140	047116	878	7652#	7655	7657	7659			
DT141	= 047116	881	7655#						
DT142	= 047116	884	7657#						
DT143	= 047116	887	7659#						

DT15	052230	624	8110#						
DT150	052672	902	8220#						
DT55	052236	722	8113#	8116	8118	8120	8122	8124	
DT56	= 052236	725	8116#						
DT57	= 052236	728	8118#						
DT60	= 052236	731	8120#						
DT61	= 052236	734	8122#						
DT62	= 052236	737	8124#						
DT63	052250	740	8126#						
DT64	052262	743	8129#	8138	8140	8142	8144	8146	8148
DT65	052272	746	8132#						
DT66	052304	749	8135#						
DT67	- 052262	752	8138#						
DT70	= 052262	755	8140#						
DT71	= 052262	758	8142#						
DT72	- 052262	761	8144#						
DT73	= 052262	764	8146#						
DT74	- 052262	767	8148#						
DT75	052320	771	8150#						
DT76	052334	774	8153#	8162					
DT77	052350	777	8156#						
EMTVEC=	000030	148#	915*	916*					
EM1	036504	588	6818#						
EM117	043260	826	7276#						
EM120	043407	829	7292#						
EM121	043622	832	7318#						
EM122	044023	835	7342#						
EM123	044153	838	7358#						
EM124	044354	841	7381#						
EM127	044562	850	7405#						
EM130	044744	853	7426#						
EM131	045016	857	7434#						
EM132	047130	860	7662#						
EM133	047267	863	7680#						
EM134	047441	866	7700#						
EM135	047607	869	7719#						
EM14	036571	621	6829#						
EM140	045243	878	7461#						
EM141	045604	881	7503#						
EM142	046144	884	7545#						
EM143	046506	887	7587#						
EM15	036630	624	6836#						
EM150	047772	902	7741#						
EM55	036700	722	6843#						
EM56	037044	725	6862#						
EM57	037211	728	6881#						
EM60	037333	731	6897#						
EM61	037457	734	6913#						
EM62	037607	737	6930#						
EM63	037735	740	6947#						
EM64	040154	743	6974#						
EM65	040353	746	6997#						
EM66	040736	749	7042#						
EM67	041020	752	7052#						
EM70	041235	755	7077#						
EM71	041513	758	7110#						





JB2	004740	1218#						
JC	= 000003	1253#						
JCDONE	005162	1271	1281	1284#				
JCERR1	005112	1269	1273#					
JC1	005064	1263	1265#					
JC2	005104	1270#						
JD	= 000004	1240	1299#					
JDDONE	005362	1329	1348	1355#				
JDERR1	005336	1323	1333	1338	1342	1346	1350#	
KA	= 000006	1465#						
KADONE	006222	1516	1534#					
KAD2	006246	1536	1540#					
KAD3	006264	1538	1541	1543#				
KAERR1	006152	1488	1520#					
KAERR2	006170	1502	1525#					
KAERR3	006206	1515	1530#					
KAFLG	006150	1474*	1518#	1523*	1528*	1533*	1535	1540
KA1	005750	1475#	1476					
KA2	005772	1477	1480#	1481				
KA3	006020	1490#	1524					
KA4	006044	1492	1494#	1495				
KA5	006072	1503#	1529					
KA6	006116	1505	1507#	1508				
KB	- 000005	1368#	1917					
KBDONE	005656	1417	1437#					
KBD2	005700	1439	1444#					
KBD3	005714	1445	1448#					
KBERR1	005606	1390	1422#					
KBERR2	005624	1403	1427#					
KBERR3	005642	1416	1432#					
KBFLG	005604	1377*	1420#	1425*	1430*	1435*	1438	1444
KBTST	003274	957#						
KB1	005416	1378#	1379					
KB11CM	001312	561#	957*	1005*	1017	1328		
KB11E	001310	559#	958*	962*	1001	1003*	1015	1023
KB11EM	001311	560#	1326					
KB2	005442	1381	1383#	1384				
KB3	005464	1392#	1426					
KB4	005510	1394	1396#	1397				
KB5	005532	1405#	1431					
KB6	005556	1407	1409#	1410				
KC	= 000011	1746#						
KCCON	007636	1755*	1775	1865*	1868*	1874#	1907	
KCDONE	007750	1862	1913#					
KCERR	007720	1852	1906#					
KCEO	007704	1841	1848	1899#	8164			
KCE1	007706	1900#	8167					
KCE2	007710	1901#	8169					
KCE3	007712	1902#	8171					
KCE4	007714	1903#	8173					
KCE5	007716	1904#	8175					
KCFLG1	007640	1756*	1860*	1876#				
KCPTR	007642	1758*	1772	1838	1855*	1856	1878#	
KCRO	007670	1830*	1847	1892#	8164			
KCR1	007672	1832*	1893#	8167				
KCR2	007674	1833*	1894#	8169				



KIPAR4= 172350	311#																
KIPAR5= 172352	312#																
KIPAR6= 172354	313#	4384*															
KIPAR7= 172356	314#	5998															
KIPDR0= 172300	285#	982*	983	985*	3587	3702	3810	3922	4033	4144	4255	4369	4790				
	4930	5071															
KIPDR1= 172302	286#																
KIPDR2= 172304	287#																
KIPDR3= 172306	288#																
KIPDR4= 172310	289#																
KIPDR5= 172312	290#																
KIPDR6= 172314	291#																
KIPDR7= 172316	292#																
KTMP1D 006610	1633#																
KTMP1E 007140	1726#																
KTMP2D 006612	1572	1634#															
KTMP2E 007142	1665	1727#															
KV = 000044	4461#																
KVDONE 023406	4494	4505#															
KVERR 023346	4473	4496#															
KV1 023330	4476	4483#															
KV2 023336	4486#																
KX = 000045	4521#																
KXDONE 023606	4555	4571#															
KXERR 023546	4547	4561#															
KX1 023510	4538	4546#															
KX2 023540	4532	4553#															
KY = 000012	1933#																
KY1 007772	1935	1937#															
KY2 010006	1938	1940#															
KZ = 000046	4586#																
KZDONE 024006	4620	4636#															
KZERR 023746	4612	4626#															
KZ1 023710	4603	4611#															
KZ2 023740	4597	4618#															
K1D 006322	1572#																
K1E 006652	1665#																
K2D 006366	1583#																
K2E 006716	1676#																
K3D 006426	1587	1594#															
K3E 006756	1680	1687#															
K4D 006472	1600	1607#															
K4E 007022	1693	1700#															
K5D 006532	1611	1618#															
K5E 007062	1704	1711#															
K6D 006602	1617	1623	1630#														
K6E 007132	1710	1716	1723#														
K7D 006614	1631	1636#															
K7E 007144	1724	1729#															
LF - 000012	49#	5703	5709														
LOADRS 177740	159#	1118	1145	1266	2145	2209	2216	2225	2294	2302	2311	2378	2386				
	2395	2465	2473	2482	2551	2559	2568	2639	2647	2656	2725	2733	2742				
	2824	2832	2841	2923	2931	2940	3022	3030	3039	3121	3129	3138	3220				
	3228	3237	3319	3327	3336	3418	3426	3435	3517	3525	3534	3642	3660				
	3667	3754	3762	3771	3866	3874	3883	3977	3985	3994	4088	4096	4105				
	4199	4207	4216	4310	4318	4327	4417	4435	4442	4500	4565	4630	4723				









MPDONE	017006	3504	3523	3529	3544#
MPERR0	016606	3471	3506#		
MP1	016566	3477	3493#		
MP2	016570	3499#			
MQ =	000043	4354#			
MQDONE	023232	4405	4440	4447#	
MQERR	023026	4386	4407#		
MQ1	023006	4401#			
MQ2	023042	4408	4411#		
MQ3	023130	4412	4426#		
MQ4	023132	4424	4428#		
MQ5	023156	4434#	4443	4444	
MQ6	023212	4432	4442#		
MR -	000034	3563#			
MRDONE	017436	3628	3665	3672#	
MRERR0	017230	3574	3630#		
MR1	017210	3624#			
MR2	017242	3632	3635#		
MR3	017332	3636	3651#		
MR4	017336	3649	3653#		
MR5	017362	3659#	3668	3670	
MR6	017416	3657	3667#		
MS =	000035	3687#			
MSDONE	020056	3741	3760	3766	3781#
MSERR0	017656	3692	3743#		
MSG1	036361	1014	6798#		
MSG2	036420	1027	6804#		
MSG3	036431	1019	6806#		
MSG4	036443	1021	6808#		
MSG5	036474	1025	6813#		
MSIZER	032552	5894	6284#		
MS1	017630	3724	3731#		
MS2	017634	3733#			
MS3	017640	3736#			
MT -	000036	3796#			
MTA101	035266	6672#			
MTA11	033534	6492#			
MTA120	035373	6685#	8164		
MTA124	035654	6727#	8186		
MTA126	035746	6739#	8193		
MTA131	036026	6750#	8202		
MTA134	036171	6771#	8211		
MTA135	036225	6777#	8215		
MTA17	033601	6500#	6523		
MTA20	033635	6509#			
MTA21	033644	6512#			
MTA43	033731	6525#			
MTA45	034004	6534#			
MTA5	033452	6482#			
MTA50	034062	6546#			
MTA77	035110	6649#	8156		
MTB120	035423	6691#	8167		
MTB126	035774	6744#	8193		
MTB131	036110	6760#	8204		
MTB135	036255	6783#	8215		
MTB17	033606	6502#			





NB1	024550	4813	4820#		
NB10	025042	4880	4891#		
NB2	024554	4822#	4863	4869	
NB3	024602	4810	4832#		
NB4	024640	4839	4846#		
NB5	024644	4848#			
NB6	024672	4836	4858#		
NB7	024716	4862	4866#		
NB8	024762	4864	4876#		
NB9	025006	4882#	4892	4894	
NC	= 000051	4916#			
NCDONE	025562	4971	4996	5030	5037#
NC1	025254	4955	4962#		
NC10	025542	5021	5032#		
NC2	025260	4964#	5004	5010	
NC3	025306	4952	4974#		
NC4	025340	4980	4987#		
NC5	025344	4989#			
NC6	025372	4977	4999#		
NC7	025416	5003	5007#		
NC8	025462	5005	5017#		
NC9	025506	5023#	5033	5035	
ND	= 000052	5057#			
NDDONE	026266	5112	5138	5172	5179#
ND1	025754	5096	5103#		
ND10	026246	5163	5174#		
ND2	025760	5105#	5146	5152	
ND3	026006	5093	5115#		
ND4	026044	5122	5129#		
ND5	026050	5131#			
ND6	026076	5119	5141#		
ND7	026122	5145	5149#		
ND8	026166	5147	5159#		
ND9	026212	5165#	5175	5177	
NMDONE	016076	3306	3325	3331	3346#
NMERRC	015676	3273	3308#		
NM1	015656	3279	3295#		
NM2	015660	3301#			
NOCNC	032504	6232	6250#		
OKSIZ	004176	1037	1071#		
PARCNT	032350	5216	5337	6209#	
PDMSG1	034110	6551#			
PDMSG2	034266	6571#			
PIRQ	= 177772	43#			
PIRQVE	- 000240	153#			
POWERM	033403	5933	6472#		
PRO	000000	75#			
PR1	= 000040	76#			
PR2	= 000100	77#			
PR3	- 000140	78#			
PR4	- 000200	79#			
PR5	- 000240	80#			
PR6	- 000300	81#			
PR7	- 000340	82#			
PS	= 177776	40#	41	907*	5972
PSW	- 177776	41#	6133*		



SKPBAD=	104426	1260	5895#											
SKPBCN=	104432	1205	1306	1375	1472	1569	1662	1753	1964	2071	2179	2255	2340	2424
		2511	2685	2775	2874	2973	3072	3171	3270	3369	3468	3570	3694	3803
		3915	4026	4137	4248	4362	4468	4528	4593	4666	4783	4923	5064	5897#
SKPBER=	104430	1261	2178	2254	2339	2423	2510	2684	2774	2873	2972	3071	3170	3269
		3368	3467	3569	3693	3802	3914	4025	4136	4247	4361	4467	4527	4592
		4665	4782	4922	5063	5896#								
SKPBHM=	104436	1376	1473	1570	1663	1754	1965	2181	2257	2342	2426	2513	2687	2777
		2876	2975	3074	3173	3272	3371	3470	3572	3696	3805	3917	4028	4139
		4250	4364	4470	4530	4595	4668	4785	4925	5066	5899#			
SKPBMN=	104434	1206	2072	2180	2256	2341	2425	2512	2686	2776	2875	2974	3073	3172
		3271	3370	3469	3571	3695	3804	3916	4027	4138	4249	4363	4469	4529
		4594	4667	4784	4924	5065	5898#							
SKRNG	032314	6156	6163	6170	6177	6185#								
SPUR	031764	1093	1110	1202	1567	1660	1751	1962	2141	2142	2155	4359	5210	5278
		5289	5331	5399	5410	6089#	6095	6123						
SR0	= 177572	190#	5978	5993*	6001*									
SR1	= 177574	191#												
SR2	= 177576	192#												
SR3	= 172516	193#	5990*											
STACK	= 001100	34#	35	36	37	912	6126							
START	003014	449	906#	5935										
STKLMT=	177774	42#												
STOP	032462	6243	6245#											
ST2	005256	1327	1330#											
SUPSTK=	000700	36#												
SWR	= 177570	44#	45	5487*	5501	5504	5512	5519	5558	5561	5568	5572	5578	6268
SW0	= 000001	110#												
SW00	= 000001	100#	110											
SW01	= 000002	99#	109											
SW02	= 000004	98#	108											
SW03	= 000010	97#	107											
SW04	= 000020	96#	106											
SW05	= 000040	95#	105											
SW06	= 000100	94#	104											
SW07	= 000200	93#	103											
SW08	= 000400	92#	102											
SW09	= 001000	91#	101											
SW1	= 000002	109#												
SW10	= 002000	90#												
SW11	= 004000	89#												
SW12	= 010000	88#												
SW13	= 020000	87#												
SW14	= 040000	86#												
SW15	= 100000	85#												
SW2	= 000004	108#												
SW3	= 000010	107#												
SW4	= 000020	106#												
SW5	= 000040	105#												
SW6	= 000100	104#												
SW7	= 000200	103#	6268											
SW8	= 000400	102#												
SW9	= 001000	101#												
SYSTID=	177764	174#												
SOMOM1=	000034	432#	1406											
SOM1	= 000030	430#	1504	1594	1676	1684	1711	1766	1868	1994	2781	2880	3177	3276











\$REG2	001160	515#												
\$REG20	001214	529#												
\$REG21	001216	530#												
\$REG22	001220	531#												
\$REG23	001222	532#												
\$REG3	001162	516#												
\$REG4	001164	517#												
\$REG5	001166	518#												
\$REG6	001170	519#												
\$REG7	001172	520#												
\$RESRE	030070	5623#	5889											
\$SAVRE	030032	5607#	5888											
\$SAVR6	031226	5914*	5920	5921*	5922*	5939#								
\$SCOPE	027360	913	5486#											
\$SETUP=	000037	422#	913	915	917	919	921	922	923	925	929	5439	5572	
\$SIZE	031230	1033	5963#											
\$SIZE X	031510	6002	6012#											
\$STUP =	177777	422#												
\$SVLAD	027576	5496	5528#											
\$SVPC -	000204	471#	476											
\$SWR	167400	11	12#	13#	21	22	23	24	25	26	27	553	554	555
		922	923	925	926	1104	1197	1252	1297	1367	1464	1561	1654	1745
		1933	1956	2064	2171	2247	2332	2416	2503	2589	2677	2767	2866	2965
		3064	3163	3262	3361	3460	3562	3686	3795	3907	4018	4129	4240	4353
		4460	4520	4585	4658	4775	4915	5056	5204	5325	5434	5440	5455	5465
		5466	5478	5479	5480	5481	5482	5487	5499	5501	5502	5508	5509	5510
		5517	5518	5519	5530	5533	5536	5545	5546	5547	5548	5549	5550	5558
		5561	5568	5572	5578	5588								
\$SWRMK-	000200	14#	27	28	5482	5483	5503	5505						
\$TAB	033450	6397	6480#											
\$TIMES	001274	553#	922*	1104*	1252*	1297*	1367*	1464*	1561*	1654*	1745*	1956*	2064*	2171*
		2247*	2332*	2416*	2503*	2589*	2677*	2767*	2866*	2965*	3064*	3163*	3262*	3361*
		3460*	3562*	3686*	3795*	3907*	4018*	4129*	4240*	4353*	4460*	4520*	4585*	4658*
		4775*	4915*	5056*	5204*	5325*	5440*	5517*	5524	5527*	5536			
\$TKB	001140	504#	1089*	6227	6250*									
\$TKS	001136	503#	910	1090	6251*									
\$TMP0	001224	533#	1109*	1201*	1257*	1303*	1372*	1469*	1566*	1659*	1750*	1961*	2069*	2176*
		2252*	2337*	2421*	2508*	2594*	2682*	2772*	2871*	2970*	3069*	3168*	3267*	3366*
		3465*	3567*	3691*	3800*	3912*	4023*	4134*	4245*	4358*	4465*	4525*	4590*	4663*
		4780*	4920*	5061*	5209*	5330*	7652	8104	8107	8110	8113	8126	8129	8132
		8135	8150	8153	8156	8164	8178	8181	8186	8190	8193	8199	8202	8211
		8215	8220											
\$TMP1	001226	534#	1141*	1273*	1988*	2004*	2145*	2793*	2892*	2991*	3090*	3189*	3288*	3387*
		3486*	4541*	4606*	6081*	6097*	6109*	8104	8107	8110	8113	8186	8220	
\$TMP10	001244	541#	1576*	1669*	8150	8156								
\$TMP11	001246	542#	1577*	1670*										
\$TMP12	001250	543#	1580*	1673*	8153	8159								
\$TMP13	001252	544#	1581*	1674*										
\$TMP14	001254	545#												
\$TMP15	001256	546#												
\$TMP16	001260	547#												
\$TMP17	001262	548#												
\$TMP2	001230	535#	1224*	1232*	1274*	1350*	1423*	1428*	1433*	1521*	1526*	1531*	1590*	1614*
		1625*	1683*	1707*	1718*	1907*	1987*	2003*	2035*	2116*	2125*	2146*	2195*	2209*
		2223*	2277*	2294*	2309*	2361*	2378*	2393*	2448*	2465*	2480*	2534*	2551*	2566*
		2622*	2639*	2654*	2708*	2725*	2740*	2792*	2807*	2824*	2839*	2891*	2906*	2923*



\$TSTNM	001102	489#	906*	1109	1201	1257	1303	1372	1469	1566	1659	1750	1961	2069
		2176	2252	2337	2421	2508	2594	2682	2772	2871	2970	3069	3168	3267
		3366	3465	3567	3691	3800	3912	4023	4134	4245	4358	4465	4525	4590
		4663	4780	4920	5061	5209	5330	5439*	5477	5506	5528*	5533	5537	5557
		5588												
\$TYPBN=	***** U	5888												
\$TYPDS	030570	5802#	5887											
\$TYPE	030126	5658#	5874	5883										
\$TYPEC	030272	5677	5684	5691	5696#	5697								
\$TYPEX	030340	5702	5704	5707#										
\$TYPOC	030366	5741#	5884											
\$TYPON	030402	5740	5743#	5886										
\$TYPOS	030342	5736#	5885											
\$XTSTR	027366	5490#												
\$GET4=	000001	5455#	5457#											
\$TRP	000002	5873#	5884	5885	5886	5887	5888	5889	5890	5892	5893	5894	5895	5896
		5897	5898	5899	5900									
\$OFILL	030565	5737*	5741*	5751	5786#									
.	060712	441#	445	447#	471	472#	474#	476#	485#	559	911	925	926	1041#
		1045#	1058#	1066#	1796	1799#	2101	2264	2267#	2349	2352#	2433	2436#	2520
		2523#	2608	2611#	2694	2697#	3726	3729#	3835	3838#	3947	3950#	4058	4061#
		4169	4172#	4280	4283#	4478	4481#	4677	4680#	4701	4704#	4815	4818#	4841
		4844#	4957	4960#	4982	4985#	5098	5101#	5124	5127#	5267	5270#	5388	5391#
		5466	5470	5536	5537	5588	5709	5856#	5917	5938	6076#	8100#	8224#	





UMAC2	1#	5275	5286	5396	5407										
UMAC3	1#	5236	5257	5357	5378										
SSCMRE	478#	513	514	515	516	517	518	519	520	521	522	523	524	525	526
		527	528	529	530	531	532								
SSCMTM	478#	533	534	535	536	537	538	539	540	541	542	543	544	545	546
		547	548	549	550	551	552								
SSESCA	1#	420#													
SSNEWT	1#	420#	1096	1188	1238	1287	1358	1450	1546	1639	1732	1915	1942	2048	2160
		2238	2323	2407	2494	2580	2668	2755	2854	2953	3052	3151	3250	3349	3448
		3674	3783	3895	4006	4117	4228	4339	4449	4509	4574	4642	4759	4899	5040
		5303													
SSSET	5874#	5884	5885	5886	5887	5888	5889	5891	5892	5893	5894	5895	5896	5897	5898
		5899													
SSSKIP	1#	420#													
.EQUAT	1#														
.HEADE	1#														
.KT11	1#														
.SETUP	1#	422													
.SWRHI	1#	16													
.SWRLO	28#														
.\$ACT1	1#	451													
.\$CATC	1#	438													
.\$CMTA	1#	478													
.\$DB2D	1#														
.\$DB2O	1#	6037													
.\$DIV	1#														
.\$EOP	1#	5427													
.\$ERRO	1#	5538													
.\$ERRT	1#														
.\$MULT	1#														
.\$POWE	1#	5901													
.\$RAND	1#														
.\$RDDE	1#														
.\$RDOC	1#														
.\$READ	1#														
.\$SAVE	1#	5589													
.\$SB2D	1#														
.\$SB2O	1#														
.\$SCOP	1#	5471													
.\$SITE	1#	5940													
.\$SUPR	1#														
.\$TRAP	1#	5858													
.\$TYPB	1#														
.\$TYPD	1#	5789													
.\$TYPE	1#	5636													
.\$TYPO	1#	5710													
.1170	1#	30													

. ABS. 060712 000

ERRORS DETECTED: 0

DSKZ:CEKBCC.BIN,DSKZ:CEKBCC.LST/CRF/SOL CEKBCC.SML,CEKBCC.P11  
 RUN-TIME: 59 86 9 SECONDS

RUN-TIME RATIO: 286/157=1.8  
CORE USED: 35K (69 PAGES)