PDP11·70/74 11/70 CACHE #1
CEKBCDO

AH-0010D-MC
FICHE 1 OF 1

0 0 3 1 3 0

.REM @

IDENTIFICATION

PRODUCT CODE: AC-0009D-MC

PRODUCT NAME: CEKBCDO 11/70 CACHE #1

DATE CREATED: MAY, 1980

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: ANTHONY VEZZA

CONTENTS

REVISION HISTORY

```
*****************************************************************************
REV D0  MODIFIED TEST 43 TO SUPPORT  CPU'S WITH >1920K MEMORY.

*****************************************************************************
```

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 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 FAILTY 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 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.

NOTE: THIS DIAGNOSTIC SUPPORTS THE PDP-11/74, AN EXPERIMENTAL, IN-HOUSE PROCESSOR.

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 TYPOUTS
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 INSTUCTION REPEATED FROM LOCATION 0 THROUGH 776 FOR THE PURPOSE OF CATCHING ANY SPURIOUS TRAP TO A VECTOR. SUCH A TRAP WILL RESULT IN A HALT AT THE TRAP VECTOR ADDRESS PLUS TWO (2).

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

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

5.2.7 MONITOR OR LOADER RESTORE - WHEN THIS PROGRAM IS FIRST STARTED IT SAVES THE CONTENTS OF THE HIGHEST 1.5 (DEC) K OF MEMORY IN THE FIRST 28K. THESE LOCATIONS USUALLY CONTAIN THE LOADER OR MONITOR OF THE SYSTEM. TO RESTORE THIS LOADER OR MONITOR THE USER NEED ONLY TYPE CONTROL C (^C) ON

THE TERMINAL AND THAT MONITOR OR LOADER WILL AUTOMATICALLY BE RESTORED. AFTER THIS IS DONE THE PROGRAM WILL HALT. NOTE THAT MANY OF THESE TESTS WIPE OUT THE ORIGINAL CONTENTS OF THAT PART OF MEMORY THEREFORE THE USER SHOULD TYPE CONTROL-C (^C) TO RESTORE THESE LOCATIONS AND AVOID HAVING TO RELOAD HIS MONITOR OR LOADER.

5.3 OPERATOR ACTION - ONLY THE POWER UP INVALIDATOR TEST IN PROGRAM 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 PROCEDING WITH THE PROGRAM'S EXECUTION WOULD PROBABLY BE FATAL, SO A HALT IS EXECUTED! NO OTHER TEST IN EITHER PROGRAM SHOULD HALT UNDER ANY NORMAL ERROR DETECTION.

6.2 ERROR RECOVERY - IF NONE OF THE ERROR PERTAINENT OPERATIONAL SWITCHES ARE BEING USED THE PROGRAM WILL EITHER RESUME THE TEST THAT MADE THE ERROR CALL OR START EXECUTION OF THE TEST FOLLOWING THE TEST DURING WHICH THE ERROR CALL WAS MADE DEPENDING ON WHETHER OR NOT THE ERROR WHICH WAS DETECTED (OR EVEN THE ERROR CALL ITSELF) WAS FATAL TO THE TEST WHICH MADE THE ERROR CALL. IF THE HALT DESCRIBED IN 6.1 ABOVE IS EVER EXECUTED 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 .$EOP.

8. MISCELLANEOUS

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

8.2 STACK POINTER - IN BOTH PROGRAMS THE STACK POINTER (R6) WILL BE INITIALIZED TO 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 OVERIDEN 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 FACILTIES. 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 IMPLIMENTED USING SW<7>. WHEN THIS SWITCH IS 0 NORMAL OPERATION IS UNDERTAKEN, BUT WHEN SW<7>=1 THEN ANY TEST WHICH MUST TURN ON THE MEMORY MANAGEMENT UNIT (THE MAPPING BOX) WILL NOT BE RUN AND CONTROL WILL BE PASSED TO THE NEXT TEST!

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

9. PROGRAM DESCRIPTION

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 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.

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

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

TEST 14 CACHE MAINTENANCE REGISTER COUNT PATTERN TEST

THIS TEST RUNS A COUNT PATTERN THROUGH THE MAINTENANCE REGISTER'S BITS 15 TO 4. THIS IS DONE TO INSURE THAT THESE BITS ARE SETABLE AND THAT THE DATA PATH TO THE REGISTERS IS VIABLE. MISSES ARE FORCED TO BOTH GROUPS SO THAT NO CACHE DATA OR ADDRESS MEMORY ERRORS SHOULD OCCUR. ALSO ANY CYCLES DONE TO MAIN MEMORY ARE INSURED, BY PROPER SELECTION OF INSTRUCTIONS, TO RETURN DATA WITH THE PARITY BITS ON SO AS TO NOT CAUSE MAIN MEMORY

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

TEST 15 CACHE MAINTENANCE AND ERROR REGISTERS TEST 1

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

TEST 16 CACHE MAINTENANCE AND ERROR REGISTERS TEST 2

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

TEST 17 CACHE MAINTENANCE AND ERROR REGISTERS TEST 3

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

TEST 20 CACHE MAINTENANCE AND ERROR REGISTERS TEST 4

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

TEST 21 CACHE MAINTENANCE AND ERROR REGISTERS TEST 5

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

TEST 22 CACHE MAINTENANCE AND ERROR REGISTERS TEST 6

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

TEST 23 CACHE MAINTENANCE AND ERROR REGISTERS TEST 7

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

TEST 24 CACHE MAINTENANCE AND ERROR REGISTERS TEST 10

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

TEST 25 CACHE MAINTENANCE AND ERROR REGISTERS TEST 11

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

NOTE: NEW MEMORY OPTIONS MAKE 2048K OF MEMORY A POSSIBILITY. IF SIZELO REG INDICATES THE PRESENCE OF MORE THAN 1920K MEMORY, THE TEST WILL BE MODIFIED SO THAT MEMORY MANAGEMENT ATTEMPTS TO ACCESS ADDRESS 17760000. THE UNIBUS MAP WILL NOT RESPOND TO THIS ADDRESS (NOR SHOULD ANY UNIBUS DEVICE) THUS GENERATING A UNIBUS TIMEOUT.

TEST 44 CACHE CONTROL REGISTER DISABLE TRAPS TEST 1

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

TEST 45 CACHE CONTROL REGISTER DISABLE TRAPS
TEST 2

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

TEST 46 CACHE CONTROL REGISTER DISABLE TRAPS
TEST 3

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

TEST 47 CACHE ERROR REGISTER LOCK UP TEST 1

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

@

```
                                .TITLE  CEKBC-D 11/70 CACHE #1
                                ;*COPYRIGHT (C) 1975, 1980
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-A5-1).
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
167400                          $SWR=167400
000200                          $SWRMK=200


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


                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100            ;;FIRST ADDRESS OF THE STACK
001100                          KERSTK= STACK           ;;KERNEL STACK
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
000003                          R3=     %3              ;;GENERAL REGISTER
```

```
                        R4=     %4              ;;GENERAL REGISTER
                        R5=     %5              ;;GENERAL REGISTER
                        R6=     %6              ;;GENERAL REGISTER
                        R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R0,R10          ;;GENERAL REGISTER
                        .EQUIV  R1,R11          ;;GENERAL REGISTER
                        .EQUIV  R2,R12          ;;GENERAL REGISTER
                        .EQUIV  R3,R13          ;;GENERAL REGISTER
                        .EQUIV  R4,R14          ;;GENERAL REGISTER
                        .EQUIV  R5,R15          ;;GENERAL REGISTER
000006                  SP=%6
                        .EQUIV  SP,KSP          ;;KERNEL STACK POINTER
                        .EQUIV  SP,SSP          ;;SUPERVISOR STACK POINTER
                        .EQUIV  SP,USP          ;;USER STACK POINTER
000007                  PC=%7

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

                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
```

```
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
                        BIT09=  1000
                        BIT08=  400
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

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

                        .SBTTL  CACHE   REGISTER DEFINITIONS


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


                        .SBTTL  CPU REGISTER DEFINITIONS
```

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



                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS


                                ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

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

                                ;*USER "I" PAGE DESCRIPTOR REGISTERS

                                UIPDR0= 177600
                                UIPDR1= 177602
                                UIPDR2= 177604
                                UIPDR3= 177606
                                UIPDR4= 177610
                                UIPDR5= 177612
                                UIPDR6= 177614
                                UIPDR7= 177616

                                ;*USER "D" PAGE DESCRIPTOR REGISTORS

                                UDPDR0= 177620
                                UDPDR1= 177622
                                UDPDR2= 177624
                                UDPDR3= 177626
                                UDPDR4= 177630
                                UDPDR5= 177632
                                UDPDR6= 177634
                                UDPDR7= 177636

                                ;*USER "I" PAGE ADDRESS REGISTERS

                                UIPAR0= 177640
                                UIPAR1= 177642
                                UIPAR2= 177644
                                UIPAR3= 177646
                                UIPAR4= 177650
                                UIPAR5= 177652
                                UIPAR6= 177654
```

```
                177656          UIPAR7= 177656
                                ;*USER 'D'' PAGE ADDRESS REGISTERS
                177660          UDPAR0= 177660
                177662          UDPAR1= 177662
                177664          UDPAR2= 177664
                177666          UDPAR3= 177666
                177670          UDPAR4= 177670
                177672          UDPAR5= 177672
                177674          UDPAR6= 177674
                177676          UDPAR7= 177676
                                ;*SUPERVISOR ''I'' PAGE DESCRIPTOR REGISTERS
                172200          SIPDR0= 172200
                172202          SIPDR1= 172202
                172204          SIPDR2= 172204
                172206          SIPDR3= 172206
                172210          SIPDR4= 172210
                172212          SIPDR5= 172212
                172214          SIPDR6= 172214
                172216          SIPDR7= 172216
                                ;*SUPERVISOR 'D'' PAGE DESCRIPTOR REGISTERS
                172220          SDPDR0= 172220
                172222          SDPDR1= 172222
                172224          SDPDR2= 172224
                172226          SDPDR3= 172226
                172230          SDPDR4= 172230
                172232          SDPDR5= 172232
                172234          SDPDR6= 172234
                172236          SDPDR7= 172236
                                ;*SUPERVISOR ''I'' PAGE ADDRESS REGISTERS
                172240          SIPAR0= 172240
                172242          SIPAR1= 172242
                172244          SIPAR2= 172244
                172246          SIPAR3= 172246
                172250          SIPAR4= 172250
                172252          SIPAR5= 172252
                172254          SIPAR6= 172254
                172256          SIPAR7= 172256
                                ;*SUPERVISOR 'D'' PAGE ADDRESS REGISTERS
                172260          SDPAR0= 172260
                172262          SDPAR1= 172262
                172264          SDPAR2= 172264
                172266          SDPAR3= 172266
                172270          SDPAR4= 172270
                172272          SDPAR5= 172272
                172274          SDPAR6= 172274
                172276          SDPAR7= 172276
```

```
                                ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

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

                                ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS

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

                                ;*KERNEL "I" PAGE ADDRESS REGISTERS

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

                                ;*KERNEL "D" PAGE ADDRESS REGISTERS

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


                                .SBTTL  UNIBUS MAP REGISTER DEFINITIONS

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

170200                          MAPL00 = 170200
```

```
  337          170202                  MAPH00 = 170202
  338          170204                  MAPL01 = 170204
  339          170206                  MAPH01 = 170206
  340          170210                  MAPL02 = 170210
  341          170212                  MAPH02 = 170212
  342          170214                  MAPL03 = 170214
  343          170216                  MAPH03 = 170216
  344          170220                  MAPL04 = 170220
  345          170222                  MAPH04 = 170222
  346          170224                  MAPL05 = 170224
  347          170226                  MAPH05 = 170226
  348          170230                  MAPL06 = 170230
  349          170232                  MAPH06 = 170232
  350          170234                  MAPL07 = 170234
  351          170236                  MAPH07 = 170236
  352          170240                  MAPL10 = 170240
  353          170242                  MAPH10 = 170242
  354          170244                  MAPL11 = 170244
  355          170246                  MAPH11 = 170246
  356          170250                  MAPL12 = 170250
  357          170252                  MAPH12 = 170252
  358          170254                  MAPL13 = 170254
  359          170256                  MAPH13 = 170256
  360          170260                  MAPL14 = 170260
  361          170262                  MAPH14 = 170262
  362          170264                  MAPL15 = 170264
  363          170266                  MAPH15 = 170266
  364          170270                  MAPL16 = 170270
  365          170272                  MAPH16 = 170272
  366          170274                  MAPL17 = 170274
  367          170276                  MAPH17 = 170276
  368          170300                  MAPL20 = 170300
  369          170302                  MAPH20 = 170302
  370          170304                  MAPL21 = 170304
  371          170306                  MAPH21 = 170306
  372          170310                  MAPL22 = 170310
  373          170312                  MAPH22 = 170312
  374          170314                  MAPL23 = 170314
  375          170316                  MAPH23 = 170316
  376          170320                  MAPL24 = 170320
  377          170320                  MAPH24 = 170320
  378          170324                  MAPL25 = 170324
  379          170326                  MAPH25 = 170326
  380          170330                  MAPL26 = 170330
  381          170332                  MAPH26 = 170332
  382          170334                  MAPL27 = 170334
  383          170336                  MAPH27 = 170336
  384          170340                  MAPL30 = 170340
  385          170342                  MAPH30 = 170342
  386          170344                  MAPL31 = 170344
  387          170346                  MAPH31 = 170346
  388          170350                  MAPL32 = 170350
  389          170352                  MAPH32 = 170352
  390          170354                  MAPL33 = 170354
  391          170356                  MAPH33 = 170356
  392          170360                  MAPL34 = 170360
```

```
                170362                  MAPH34 = 170362
                170364                  MAPL35 = 170364
                170366                  MAPH35 = 170366
                170370                  MAPL36 = 170370
                170372                  MAPH36 = 170372
                170374                  MAPL37 = 170374
                170376                  MAPH37 = 170376
                                        .EQUIV  MAPL00,MAPL0
                                        .EQUIV  MAPH00,MAPH0
                                        .EQUIV  MAPL01,MAPL1
                                        .EQUIV  MAPH01,MAPH1
                                        .EQUIV  MAPL02,MAPL2
                                        .EQUIV  MAPH02,MAPH2
                                        .EQUIV  MAPL03,MAPL3
                                        .EQUIV  MAPH03,MAPH3
                                        .EQUIV  MAPL04,MAPL4
                                        .EQUIV  MAPH04,MAPH4
                                        .EQUIV  MAPL05,MAPL5
                                        .EQUIV  MAPH05,MAPH5
                                        .EQUIV  MAPL06,MAPL6
                                        .EQUIV  MAPH06,MAPH6
                                        .EQUIV  MAPL07,MAPL7
                                        .EQUIV  MAPH07,MAPH7


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

                                        .SBTTL  TRAP CATCHER

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

                                        .SBTTL  STARTING ADDRESS(ES)
                000200                          .=200

000200  000137  003014                          JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

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

                                .SBTTL          ACT11 HOOKS

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

            000204                      $SVPC=.                 ;;SAVE LOCATION COUNTER
            000046                      .=46                    ;;SET LOCATION COUNTER
000046      027314                      .WORD   $ENDAD          ;;SET LOC.46 TO ADDRESS $ENDAD
            000052                      .=52                    ;;SET LOCATION COUNTER
000052      000000                      .WORD   0               ;;SET LOC.52 TO ZERO
            000204                      .=$SVPC                 ;; RESTORE LOCATION COUNTER
```

```
                                ;;*****************************************************************
                                .SBTTL  COMMON TAGS

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

                001100                  .=1100

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

```
    533  001226  000000                  $TMP1:   .WORD   0                ;;USER DEFINED
    534  001230  000000                  $TMP2:   .WORD   0                ;;USER DEFINED
    535  001232  000000                  $TMP3:   .WORD   0                ;;USER DEFINED
    536  001234  000000                  $TMP4:   .WORD   0                ;;USER DEFINED
    537  001236  000000                  $TMP5:   .WORD   0                ;;USER DEFINED
    538  001240  000000                  $TMP6:   .WORD   0                ;;USER DEFINED
    539  001242  000000                  $TMP7:   .WORD   0                ;;USER DEFINED
    540  001244  000000                  $TMP10:  .WORD   0                ;;USER DEFINED
    541  001246  000000                  $TMP11:  .WORD   0                ;;USER DEFINED
    542  001250  000000                  $TMP12:  .WORD   0                ;;USER DEFINED
    543  001252  000000                  $TMP13:  .WORD   0                ;;USER DEFINED
    544  001254  000000                  $TMP14:  .WORD   0                ;;USER DEFINED
    545  001256  000000                  $TMP15:  .WORD   0                ;;USER DEFINED
    546  001260  000000                  $TMP16:  .WORD   0                ;;USER DEFINED
    547  001262  000000                  $TMP17:  .WORD   0                ;;USER DEFINED
    548  001264  000000                  $TMP20:  .WORD   0                ;;USER DEFINED
    549  001266  000000                  $TMP21:  .WORD   0                ;;USER DEFINED
    550  001270  000000                  $TMP22:  .WORD   0                ;;USER DEFINED
    551  001272  000000                  $TMP23:  .WORD   0                ;;USER DEFINED
    552  001274  000000                  $TIMES:  0                        ;;MAX. NUMBER OF ITERATIONS
    553  001276  000000                  $ESCAPE:0                         ;;ESCAPE ON ERROR
    554  001300  177607  000377          $BELL:   .ASCIZ  <207><377><377>  ;;CODE FOR BELL
    555  001304     077                  $QUES:   .ASCII  /?/              ;;QUESTION MARK
    556  001305     015                  $CRLF:   .ASCII  <15>             ;;CARRIAGE RETURN
    557  001306  000012                  $LF:     .ASCIZ  <12>             ;;LINE FEED
    558  001310     000                  KB11E:   .BYTE   0        ;1174 WITHOUT MP CACHE FLAG
    559  001311     000                  KB11EM:  .BYTE   0        ;1174 WITH MP CACHE FLAG
    560  001312     000                  KB11CM:  .BYTE   0        ;KB11CM FLAG (1170 WITH MP MODS)
    561  001313     000                  CISP:    .BYTE   0        ;CISP OPTION PRESENT FLAG
    562
    563                                  ;OPCODE FOR MFPT INSTRUCTION (AVAILABLE ON KB11-E AND KB11-EM ONLY)
    564          000007                          MFPT=7
```

```
                                ;;*****************************************************************************
                                .SBTTL  ERROR POINTER TABLE

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

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

001314                          $ERRTB:


                                ;ERROR TABLE FOR ERROR TYPE OUT:
                                ;ITEM 1
001314  036474  050046  052172          .WORD   EM1,DH1,DT1,DF1
001322  051775
                                ;ITEM 0
001324  000000  000000  000000          .WORD   0,0,0,0
001332  000000
                                ;ITEM 0
001334  000000  000000  000000          .WORD   0,0,0,0
001342  000000
                                ;ITEM 0
001344  000000  000000  000000          .WORD   0,0,0,0
001352  000000
                                ;ITEM 0
001354  000000  000000  000000          .WORD   0,0,0,0
001362  000000
                                ;ITEM 0
001364  000000  000000  000000          .WORD   0,0,0,0
001372  000000
                                ;ITEM 0
001374  000000  000000  000000          .WORD   0,0,0,0
001402  000000
                                ;ITEM 0
001404  000000  000000  000000          .WORD   0,0,0,0
001412  000000
                                ;ITEM 0
001414  000000  000000  000000          .WORD   0,0,0,0
001422  000000
                                ;ITEM 0
001424  000000  000000  000000          .WORD   0,0,0,0
001432  000000
                                ;ITEM 0
001434  000000  000000  000000          .WORD   0,0,0,0
001442  000000
                                ;ITEM 14
001444  036561  050121  052204          .WORD   EM14,DH14,DT14,DF14
```

```
  621  001452  052001
  622                                   ;ITEM 15
  623  001454  036620  050214  052220                .WORD   EM15,DH15,DT15,DF15
  624  001462  052006
  625                                   ;ITEM 0
  626  001464  000000  000000  000000                .WORD   0,0,0,0
  627  001472  000000
  628                                   ;ITEM 0
  629  001474  000000  000000  000000                .WORD   0,0,0,0
  630  001502  000000
  631                                   ;ITEM 0
  632  001504  000000  000000  000000                .WORD   0,0,0,0
  633  001512  000000
  634                                   ;ITEM 0
  635  001514  000000  000000  000000                .WORD   0,0,0,0
  636  001522  000000
  637                                   ;ITEM 0
  638  001524  000000  000000  000000                .WORD   0,0,0,0
  639  001532  000000
  640                                   ;ITEM 0
  641  001534  000000  000000  000000                .WORD   0,0,0,0
  642  001542  000000
  643                                   ;ITEM 0
  644  001544  000000  000000  000000                .WORD   0,0,0,0
  645  001552  000000
  646                                   ;ITEM 0
  647  001554  000000  000000  000000                .WORD   0,0,0,0
  648  001562  000000
  649                                   ;ITEM 0
  650  001564  000000  000000  000000                .WORD   0,0,0,0
  651  001572  000000
  652                                   ;ITEM 0
  653  001574  000000  000000  000000                .WORD   0,0,0,0
  654  001602  000000
  655                                   ;ITEM 0
  656  001604  000000  000000  000000                .WORD   0,0,0,0
  657  001612  000000
  658
  659                                   ;ITEM 0
  660  001614  000000  000000  000000                .WORD   0,0,0,0
  661  001622  000000
  662                                   ;ITEM 0
  663  001624  000000  000000  000000                .WORD   0,0,0,0
  664  001632  000000
  665                                   ;ITEM 0
  666  001634  000000  000000  000000                .WORD   0,0,0,0
  667  001642  000000
  668                                   ;ITEM 0
  669  001644  000000  000000  000000                .WORD   0,0,0,0
  670  001652  000000
  671                                   ;ITEM 0
  672  001654  000000  000000  000000                .WORD   0,0,0,0
  673  001662  000000
  674                                   ;ITEM 0
  675  001664  000000  000000  000000                .WORD   0,0,0,0
  676  001672  000000
```

```
                                        ;ITEM 0
001674  000000  000000  000000                .WORD   0,0,0,0
001702  000000
                                        ;ITEM 0
001704  000000  000000  000000                .WORD   0,0,0,0
001712  000000
                                        ;ITEM 0
001714  000000  000000  000000                .WORD   0,0,0,0
001722  000000
                                        ;ITEM 0
001724  000000  000000  000000                .WORD   0,0,0,0
001732  000000
                                        ;ITEM 0
001734  000000  000000  000000                .WORD   0,0,0,0
001742  000000
                                        ;ITEM 0
001744  000000  000000  000000                .WORD   0,0,0,0
001752  000000
                                        ;ITEM 0
001754  000000  000000  000000                .WORD   0,0,0,0
001762  000000
                                        ;ITEM 0
001764  000000  000000  000000                .WORD   0,0,0,0
001772  000000
                                        ;ITEM 0
001774  000000  000000  000000                .WORD   0,0,0,0
002002  000000
                                        ;ITEM 0
002004  000000  000000  000000                .WORD   0,0,0,0
002012  000000
                                        ;ITEM 0
002014  000000  000000  000000                .WORD   0,0,0,0
002022  000000
                                        ;ITEM 0
002024  000000  000000  000000                .WORD   0,0,0,0
002032  000000
                                        ;ITEM 0
002034  000000  000000  000000                .WORD   0,0,0,0
002042  000000
                                        ;ITEM 0
002044  000000  000000  000000                .WORD   0,0,0,0
002052  000000

                                        ;ITEM 55
002054  036670  050240  052226                .WORD   EM55,DH55,DT55,DF55
002062  052010
                                        ;ITEM 56
002064  037034  050240  052226                .WORD   EM56,DH56,DT56,DF56
002072  052010
                                        ;ITEM 57
002074  037201  050240  052226                .WORD   EM57,DH57,DT57,DF57
002102  052010
                                        ;ITEM 60
002104  037323  050240  052226                .WORD   EM60,DH60,DT60,DF60
002112  052010
                                        ;ITEM 61
```

```
733  002114  037447  050240  052226          .WORD   EM61,DH61,DT61,DF61
734  002122  052010
735                                  ;ITEM 62
736  002124  037577  050240  052226          .WORD   EM62,DH62,DT62,DF62
737  002132  052010
738                                  ;ITEM 63
739  002134  037725  050315  052240          .WORD   EM63,DH63,DT63,DF63
740  002142  052014
741                                  ;ITEM 64
742  002144  040144  050417  052252          .WORD   EM64,DH64,DT64,DF64
743  002152  052014
744                                  ;ITEM 65
745  002154  040343  050472  052262          .WORD   EM65,DH65,DT65,DF65
746  002162  052014
747                                  ;ITEM 66
748  002164  040726  050574  052274          .WORD   EM66,DH66,DT66,DF66
749  002172  052014
750                                  ;ITEM 67
751  002174  041010  050647  052252          .WORD   EM67,DH67,DT67,DF67
752  002202  052014
753                                  ;ITEM 70
754  002204  041225  050647  052252          .WORD   EM70,DH70,DT70,DF70
755  002212  052014
756                                  ;ITEM 71
757  002214  041503  050647  052252          .WORD   EM71,DH71,DT71,DF71
758  002222  052014
759                                  ;ITEM 72
760  002224  041761  050647  052252          .WORD   EM72,DH72,DT72,DF72
761  002232  052014
762                                  ;ITEM 73
763  002234  042203  050647  052252          .WORD   EM73,DH73,DT73,DF73
764  002242  052014
765                                  ;ITEM 74
766  002244  042467  050647  052252          .WORD   EM74,DH74,DT74,DF74
767  002252  052014
768
769                                  ;ITEM 75
770  002254  042753  050744  052310          .WORD   EM75,DH75,DT75,DF75
771  002262  052021
772                                  ;ITEM 76
773  002264  042753  050744  052324          .WORD   EM76,DH76,DT76,DF76
774  002272  052021
775                                  ;ITEM 77
776  002274  043112  051041  052340          .WORD   EM77,DH77,DT77,DF77
777  002302  052026
778                                  ;ITEM 0
779  002304  000000  000000  000000          .WORD   0,0,0,0
780  002312  000000
781                                  ;ITEM 0
782  002314  000000  000000  000000          .WORD   0,0,0,0
783  002322  000000
784                                  ;ITEM 0
785  002324  000000  000000  000000          .WORD   0,0,0,0
786  002332  000000
787                                  ;ITEM 0
788  002334  000000  000000  000000          .WORD   0,0,0,0
```

```
 789  002342  000000
 790                                  ;ITEM 0
 791  002344  000000  000000  000000          .WORD   0,0,0,0
 792  002352  000000
 793                                  ;ITEM 0
 794  002354  000000  000000  000000          .WORD   0,0,0,0
 795  002362  000000
 796                                  ;ITEM 0
 797  002364  000000  000000  000000          .WORD   0,0,0,0
 798  002372  000000
 799                                  ;ITEM 0
 800  002374  000000  000000  000000          .WORD   0,0,0,0
 801  002402  000000
 802                                  ;ITEM 0
 803  002404  000000  000000  000000          .WORD   0,0,0,0
 804  002412  000000
 805                                  ;ITEM 0
 806  002414  000000  000000  000000          .WORD   0,0,0,0
 807  002422  000000
 808                                  ;ITEM 0
 809  002424  000000  000000  000000          .WORD   0,0,0,0
 810  002432  000000
 811
 812                                  ;ITEM 0
 813  002434  000000  000000  000000          .WORD   0,0,0,0
 814  002442  000000
 815                                  ;ITEM 0
 816  002444  000000  000000  000000          .WORD   0,0,0,0
 817  002452  000000
 818                                  ;ITEM 0
 819  002454  000000  000000  000000          .WORD   0,0,0,0
 820  002462  000000
 821                                  ;ITEM 0
 822  002464  000000  000000  000000          .WORD   0,0,0,0
 823  002472  000000
 824                                  ;ITEM 117
 825  002474  043250  050744  052324          .WORD   EM117,DH117,DT117,DF117
 826  002502  052021
 827                                  ;ITEM 120
 828  002504  043377  051065  052366          .WORD   EM120,DH120,DT120,DF120
 829  002512  052040
 830                                  ;ITEM 121
 831  002514  043612  051141  052456          .WORD   EM121,DH121,DT121,DF121
 832  002522  052073
 833                                  ;ITEM 122
 834  002524  044013  051203  052470          .WORD   EM122,DH122,DT122,DF122
 835  002532  052077
 836                                  ;ITEM 123
 837  002534  044143  051265  052470          .WORD   EM123,DH123,DT123,DF123
 838  002542  052077
 839                                  ;ITEM 124
 840  002544  044344  050121  052502          .WORD   EM124,DH124,DT124,DF124
 841  002552  052103
 842                                  ;ITEM 0
 843  002554  000000  000000  000000          .WORD   0,0,0,0
 844  002562  000000
```

```
 845                                            ;ITEM 0
 846  002564  000000  000000  000000                    .WORD   0,0,0,0
 847  002572  000000
 848                                            ;ITEM 127
 849  002574  044552  051435  052522                    .WORD   EM127,DH127,DT127,DF127
 850  002602  052127
 851                                            ;ITEM 130
 852  002604  044734  051477  052554                    .WORD   EM130,DH130,DT130,DF130
 853  002612  052113
 854
 855                                            ;ITEM 131
 856  002614  045006  051555  052566                    .WORD   EM131,DH131,DT131,DF131
 857  002622  052132
 858                                            ;ITEM 132
 859  002624  047120  051325  052522                    .WORD   EM132,DH132,DT132,DF132
 860  002632  052113
 861                                            ;ITEM 133
 862  002634  047257  051362  052532                    .WORD   EM133,DH133,DT133,DF133
 863  002642  052117
 864                                            ;ITEM 134
 865  002644  047431  051634  052614                    .WORD   EM134,DH134,DT134,DF134
 866  002652  052144
 867                                            ;ITEM 135
 868  002654  047577  051041  052634                    .WORD   EM135,DH135,DT135,DF135
 869  002662  052153
 870                                            ;ITEM 0
 871  002664  000000  000000  000000                    .WORD   0,0,0,0
 872  002672  000000
 873                                            ;ITEM 0
 874  002674  000000  000000  000000                    .WORD   0,0,0,0
 875  002702  000000
 876                                            ;ITEM 140
 877  002704  045233  047037  047106                    .WORD   EM140,DH140,DT140,DF140
 878  002712  047102
 879                                            ;ITEM 141
 880  002714  045574  047037  047106                    .WORD   EM141,DH141,DT141,DF141
 881  002722  047102
 882                                            ;ITEM 142
 883  002724  046134  047037  047106                    .WORD   EM142,DH142,DT142,DF142
 884  002732  047102
 885                                            ;ITEM 143
 886  002734  046476  047037  047106                    .WORD   EM143,DH143,DT143,DF143
 887  002742  047102
 888                                            ;ITEM 0
 889  002744  000000  000000  000000                    .WORD   0,0,0,0
 890  002752  000000
 891                                            ;ITEM 0
 892  002754  000000  000000  000000                    .WORD   0,0,0,0
 893  002762  000000
 894                                            ;ITEM 0
 895  002764  000000  000000  000000                    .WORD   0,0,0,0
 896  002772  000000
 897                                            ;ITEM 0
 898  002774  000000  000000  000000                    .WORD   0,0,0,0
 899  003002  000000
 900                                            ;ITEM 150
```

```
  901  003004  047762  051711  052662          .WORD   EM150,DH150,DT150,DF150
  902  003012  052165
  903
  904
  905  003014  005037  001102          START:  CLR     $TSTNM
  906  003020  012737  000340  177776          MOV     #340,@#PS       ;;LOCK OUT ALL INTERRUPTS
  907  003026  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
  908  003032  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
  909  003034  022706  001136                  CMP     #$TKS,R6        ;;DONE?
  910  003040  001374                          BNE     .-6             ;;LOOP BACK IF NO
  911  003042  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
  912  003046  012737  027350  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
  913  003054  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
  914  003062  012737  027632  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
  915  003070  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
  916  003076  012737  031004  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
  917  003104  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
  918  003112  012737  031064  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
  919  003120  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
  920  003126  013737  027244  027236          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
  921  003134  005037  001274                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
  922  003140  005037  001276                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
  923  003144  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
  924  003152  012737  003152  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
  925  003160  012737  003160  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
  926  003166  005227  177777                  INC     #-1             ;;FIRST TIME?
  927  003172  001024                          BNE     64$             ;;BRANCH IF NO
  928  003174  022737  027314  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
  929  003202  001420                          BEQ     64$             ;;BRANCH IF YES
  930  003204  104400  003212                  TYPE    ,65$            ;;TYPE ASCIZ STRING
  931  003210  000415                          BR      64$             ;;GET OVER THE ASCIZ
  932                                ;;65$:   .ASCIZ  <CRLF>'CEKBC-D  11/70 CACHE #1'<CRLF>
  933  003244                       64$:
  934                               ;THIS ROUTINE SAVES THE TOP 1500 (DEC) WORDS OF THE FIRST 28K OF
  935                               ;MEMORY. THESE LOCATIONS SHOULD CONTAIN EITHER THE MONITOR OR THE
  936                               ;LOADER WHICH LOADED THE PROGRAM. NOTE THAT TO RESTORE THIS PART
  937                               ;OF CORE, THAT IS TO RESTORE THE LOADER OR MONITOR, ALL THE USER
  938                               ;MUST DO IS TYPE ^C (CONTROL-C), WHILE THIS PROGRAM IS RUNNING.
  939                               ;THIS WILL AUTOMATICALLY RESTORE THE TOP PART OF MEMORY TO ITS STATE
  940                               ;BEFORE THIS PROGRAM WAS STARTED! AFTER THE MONITOR (OR LOADER) HAS BEEN
  941                               ;RESTORED THIS PROGRAM WILL HALT.
  942
  943                               ;;
  944                               ;;*** TEST FOR VARIOUS KB11 PROCESSORS ***
  945                               ;;
  946                               ;;*THIS ROUTINE POLES THE RESULTS OF ATTEMPTS TO SET TO ONE
  947                               ;;*CERTAIN CRITICAL BITS THAT ARE KNOWN TO BE OPERATIVE ON A KB11CM,
  948                               ;;*OR KB11EM PROCESSOR. IF TWO OUT OF FOUR OF THE TESTS ARE
  949                               ;;*POSITIVE THEN THE KB11CM OR KB11EM FLAG IS SET.IF LESS THAN TWO OF THE
  950                               ;;*TESTS ARE POSITIVE THEN THE KB11E FLAG OR NO FLAG IS SET. THE DETERMINATION
  951                               ;;*OF WHICH PAIR IS VALID IS BASED ON THE RESULTS OF EXECUTING AN MFPT OPCODE
  952                               ;;*(OPCODE 7).  IF THIS INSTRUCTION TRAPS THIS IS AN KB11CM OR
  953                               ;;*A PLAIN 1170 (KB11-B OR KB11-C).  IF THE INSTRUCTION DOES NOT TRAP THEN
  954                               ;;*THIS IS A KB11-E OR KB11-EM.
  955                               ;
  956  003244  105037  001312  KBTST:  CLRB    @#KB11CM        ;RESET THE MP FLAG
```

```
                                                        CLR     @#KB11E         ;CLEAR KB11E AND KB11EM FLAGS
                                                        MOV     #MFPTTR,@#RESVEC ;SET UP TRAP ADDRESS FOR MFPT AT RESERV VECTOR
                                                        MFPT                    ;EXECUTE MFPT. WILL TRAP ON 1170 (KB11B/C) OR
                                                                                ;KB11CM
                                                        MOV     #1,@#KB11E      ;HERE IF KB11E OR KB11EM. SET FLAG
                                                T1:     CLR     @#MAINT         ;CLEAR THE MAINTENANCE REGISTER
                                                        CLR     R5              ;RESET THE TEST COUNTER
                                                        MOV     #CONTRL,R0      ;GET THE ADDRESS OF...
                                                        MOV     #MAINT,R1       ;CCR,MAINT,AND MAPH00...
                                                        MOV     #MAPH00,R2      ;AND PLACE IN R0-R2
                                                        BIS     #BIT14,(R0)     ;TRY TO SET IVSS BIT
                                                        BIT     #BIT14,(R0)     ;DID IT SET?
                                                        BEQ     T2              ;NO,GO TO NEXT TEST
                                                        BIC     #BIT14,(R0)     ;CLEAR IT.
                                                        INC     R5              ;TEST IS POSITIVE
                                                T2:     BIS     #BIT0,(R1)      ;SET EDMA IN MAINT REGISTER
                                                        BIT     #BIT0,(R1)
                                                        BEQ     T3
                                                        BIS     #BIT11,(R0)     ;TRY TO SET DMMA IN CCR
                                                        BIT     #BIT11,(R0)
                                                        BEQ     T3
                                                        BIC     #BIT11,(R0)
                                                        INC     R5
                                                T3:     BIC     #BIT0,(R1)      ;MAKE SURE EDMA IS CLEAR
                                                        BIS     #BIT15,KIPDR0   ;TRY TO SET BYP ON A PDR
                                                        BIT     #BIT15,KIPDR0
                                                        BEQ     T4
                                                        BIC     #BIT15,KIPDR0
                                                        INC     R5
                                                T4:     BIS     #BIT15,(R2)     ;TRY TO SET BYP ON UNIBUS MAP
                                                        BIT     #BIT15,(R2)
                                                        BEQ     T.END
                                                        BIC     #BIT15,(R2)
                                                        INC     R5
                                                T.END:  CMP     #2,R5           ;IS THE RESULT OF THE TEST >=2
                                                        BHI     2$              ;NO,THIS IT A KB11E OR KB11-B/C (11/70)
                                                        CLR     R0
                                                        CLR     @#CONTRL
                                                3$:     MOV     @#CONTRL,R1
                                                        BEQ     4$
                                                        INC     R0
                                                        BNE     3$
                                                4$:
                                                        TST     @#KB11E         ;IS IS A KB11-E OR KB11-EM?
                                                        BEQ     1$              ;BR IF NEITHER. MUST BE KB11CM
                                                        MOV     #BIT8,@#KB11E   ;SET UPPER BYTE (KB11-EM)
                                                        BR      2$              ;DONE
                                                1$:     INCB    @#KB11CM        ;YES, FLAG THIS AS A MODIFIED PROCESSOR
                                                2$:     BR      ENDKB           ;DONE DETERMINING WHICH CPU

                                                MFPTTR:                         ;HERE IF MFPT TRAPPED. SEE IF 1170 OR KB11CM
                                                        MOV     #T1,(SP)        ;SET UP RETURN ADDRESS FOR RTI
                                                        RTI                     ;RETURN
                                                ENDKB:
                                                        INC     #-1             ;FIRST TIME?
                                                        BNE     100$            ;BR IF NO
```

```
 1013  003526  104400  036351                  TYPE    ,MSG1           ;<15><12>CPU UNDER TEST FOUND TO BE A
 1014  003532  005737  001310                  TST     @#KB11E         ;IS THIS A KB11-E OR KB11-EM?
 1015  003536  001011                          BNE     101$            ;BR IF EITHER ONE
 1016  003540  105737  001312                  TSTB    @#KB11CM        ;IS IT A KB11CM
 1017  003544  001003                          BNE     1$              ;BR IF IT IS
 1018  003546  104400  036421                  TYPE    ,MSG3           ;KB11-B/C<15><12>
 1019  003552  000413                          BR      100$            ;SKIP OTHER MESSAGE
 1020  003554  104400  036433          1$:     TYPE    ,MSG4           ;KB-CM11<15><12>
 1021  003560  000410                          BR      100$            ;SKIP CISP MESSAGE
 1022  003562  105737  001310          101$:   TSTB    @#KB11E         ;IS IT A KB11-E?
 1023  003566  001403                          BEQ     102$            ;BR IF NOT. MUST BE KB11-EM
 1024  003570  104400  036464                  TYPE    ,MSG5           ;KB11-E<15><12>
 1025  003574  000402                          BR      100$            ;SKIP KB11-EM MESSAGE
 1026  003576  104400  036410          102$:   TYPE    ,MSG2           ;KB11-EM<15><12>
 1027  003602                          100$:
 1028                                  ;;**************************************************************************
 1029                                          ;SIZE MEMORY AND COMPARE IT WITH THE SYSTEM SIZE REGISTER
 1030                                          ;PRINT A WARNING IF THEY DISAGREE.
 1031  003602  052737  000200  031266          BIS     #BIT07,$KT11
 1032  003610  004737  031220                  JSR     PC,$SIZE
 1033  003614  062737  000037  031604          ADD     #37,$LSTBK              ;ADJUST THE SIZE FOR PROPER
 1034                                                                          ;COMPARISON TO SIZE REGISTER
 1035  003622  023737  177760  031604          CMP     @#SIZELO,$LSTBK         ;SIZE REGISTER EQUAL TO ACTUAL SIZE?
 1036  003630  001546                          BEQ     OKSIZ
 1037  003632  104400  003640                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 1038  003636  000433                          BR      64$             ;;GET OVER THE ASCIZ
 1039                                  ;;65$:  .ASCIZ  <15><12>/WARNING- THE SIZE OF MEMORY IS DIFFERENT FROM THAT/
 1040  003726                          64$:
 1041  003726  104400  003734                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 1042  003732  000425                          BR      66$             ;;GET OVER THE ASCIZ
 1043                                  ;;67$:  .ASCIZ  <15><12>/INDICATED BY THE SYSTEM SIZE REGISTER./
 1044  004006                          66$:
 1045  004006  104400  004014                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 1046  004012  000421                          BR      68$             ;;GET OVER THE ASCIZ
 1047                                  ;;69$:  .ASCIZ  <15><12>/     SIZEHI  SIZELO      ACTUAL/
 1048  004056                          68$:
 1049  004056  104400  001305                  TYPE    ,$CRLF
 1050  004062  013746  177762                  MOV     @#SIZEHI,-(SP)  ;;SAVE @#SIZEHI FOR TYPEOUT
 1051  004066  104404                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 1052  004070     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
 1053  004071     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 1054  004072  104400  004100                  TYPE    ,71$            ;;TYPE ASCIZ STRING
 1055  004076  000404                          BR      70$             ;;GET OVER THE ASCIZ
 1056                                  ;;71$:  .ASCIZ  /       /
 1057  004110                          70$:
 1058  004110  013746  177760                  MOV     @#SIZELO,-(SP)  ;;SAVE @#SIZELO FOR TYPEOUT
 1059  004114  104404                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 1060  004116     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
 1061  004117     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 1062  004120  104400  004126                  TYPE    ,73$            ;;TYPE ASCIZ STRING
 1063  004124  000404                          BR      72$             ;;GET OVER THE ASCIZ
 1064                                  ;;73$:  .ASCIZ  /       /
 1065  004136                          72$:
 1066  004136  013746  031604                  MOV     $LSTBK,-(SP)    ;;SAVE $LSTBK FOR TYPEOUT
 1067  004142  104404                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 1068  004144     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
```

```
1069  004145     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
1070  004146                          OKSIZ:
1071
1072                                  ;;*************************************************************************
1073  004146  005237  032516          LOOP:   INC     MONF            ;INCREMENT THE FLAG WHICH INDICATES
1074  004152  001013                          BNE     TOP             ;WHETHER OR NOT THE TOP OF MEMORY
1075                                                                  ;IN THE FIRST 28K HAS BEEN SAVED.
1076  004154  013737  000060  032514          MOV     @#TKVEC,MONTTY  ;SAVE THE INITIAL CONTENTS OF THE TTY KEYBOARD
1077                                                                  ;VECTOR.
1078  004162  012700  002734                  MOV     #^D1500,R0      ;IF NOT THEN SAVE IT.
1079  004166  012701  052700                  MOV     #BOTTOM+4,R1    ;SAVE IT AT THE BOTTOM OF THIS PROGRAM.
1080  004172  012702  160000                  MOV     #160000,R2      ;GET THE ADDRESS OF THE END OF THE MONITOR.
1081  004176  014221                  1$:     MOV     -(R2),(R1)+     ;SAVE 1500 (DEC) LOCATIONS (WORDS)
1082  004200  077002                          SOB     R0,1$
1083  004202  012737  000044  177770  TOP:    MOV     #44,@#177770    ;SET TO SYNC SCOPE (OSCILLOSCOPE)
1084                                                                  ;ON A NOP INSTRUCTION.
1085
1086  004210  012737  032362  000060          MOV     #RESMON,@#TKVEC ;SET UP THE KEYBORD INTERRUPT VECTOR.
1087  004216  012737  000340  000062          MOV     #340,@#TKVEC+2
1088  004224  005077  174710                  CLR     @$TKB           ;MAKE SURE THE BUFFER IS CLEAR
1089  004230  152777  000100  174700          BISB    #BIT6,@$TKS     ;TURN ON INTERRUPT ENABLE FOR THE KEYBOARD.
1090
1091  004236  012737  031726  000004          MOV     #CPSPUR,@#ERRVEC        ;SET UP FOR UNEXPECTED ERRORS.
1092  004244  012737  031754  000114          MOV     #SPUR,@#CACHVEC
1093
1094
1095                                  ;;*************************************************************************
1096                                  ;*TEST 1         CACHE REGISTERS RESPONSE TEST
1097                                  ;*
1098                                  ;*REFERENCE EACH CACHE REGISTER MAKING SURE SUCH
1099                                  ;*REFERENCES DO NOT TIME OUT.
1100                                  ;*
1101                                  ;;*************************************************************************
1102  004252  000004                  TST1:   SCOPE
1103  004254  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1104           000001                 JA=$TN-1
1105                                                                  ;SET THE SKAD REGISTER
1106  004262  012737  004626  032100          MOV     #TST2,SKAD      ;IN CASE THE TEST ABORTS.
1107
1108  004270  113737  001102  001224          MOVB    $TSTNM,$TMPO
1109  004276  012737  031754  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
1110  004304  012701  032310                  MOV     #LOAFLG,R1      ;CLEAR THE REGISTER FLAGS
1111  004310  012700  000014                  MOV     #14,R0
1112  004314  005021                  64$:    CLR     (R1)+
1113  004316  077002                          SOB     R0,64$
1114  004320  013737  000004  004376          MOV     @#ERRVEC,JATMP  ;SAVE THE OLD CONTENTS OF VECTOR ERRVEC.
1115  004326  012737  004400  000004          MOV     #JAERR,@#ERRVEC ;SET UP THE TIME OUT
1116                                                                  ;VECTOR
1117  004334  012700  177740                  MOV     #LOADRS,R0
1118  004340  012737  004346  001110          MOV     #JA1,$LPERR
1119
1120  004346  000240                  JA1:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
1121  004350  005710                          TST     (R0)            ;REFERENCE EACH CACHE REGISTER
1122                                                                  ;MAKING SURE EACH DOESN'T TIME OUT.
1123
1124  004352  062700  000002          JA2:    ADD     #2,R0
```

```
1125  004356  020027  177752          CMP     R0,#HITMIS
1126  004362  101771                  BLOS    JA1
1127
1128  004364  013737  004376  000004  JA3:    MOV     JATMP,@#ERRVEC  ;RESET THE CPU TRAP VECTOR.
1129  004372  000137  004622          JMP     JADONE
1130
1131  004376  000000          JATMP:  .WORD   0               ;SAVE THE OLD CONTENTS OF
1132                                                          ;VECTOR ERRVEC HERE.
1133
1134  004400  032737  000020  177766  JAERR:  BIT     #20,@#CPUERR
1135  004406  001005                  BNE     JAERR1          ;MAKE SURE THE ERROR
1136  004410  013737  004376  000004  JAERR0: MOV     JATMP,@#ERRVEC  ;IF NOT RESET VECTOR ERRVEC AND GO TO
1137  004416  000177  173362          JMP     @ERRVEC         ;THE ROUTINE WHICH HANDLES CPU ERRORS.
1138  004422  021627  004352          JAERR1: CMP     (SP),#JA2       ;OTHERWISE REPORT THE FACT THAT A CACHE
1139  004426  001370                  BNE     JAERR0          ;REGISTER REFERENCE TIMED OUT!
1140  004430  012637  001226          MOV     (SP)+,$TMP1
1141  004434  005726                  TST     (SP)+
1142  004436  010037  001232          MOV     R0,$TMP3
1143  004442  012737  000077  001234  MOV     #77,$TMP4
1144  004450  020027  177740          CMP     R0,#LOADRS
1145  004454  001005                  BNE     JAERR2
1146  004456  012737  177777  032310  MOV     #-1,LOAFLG
1147  004464  104055          1$:     ERROR   55              ;CACHE REGISTER RESPONSE TEST FAILED
1148  004466  000451                  BR      JAERR9
1149
1150  004470  020027  177742          JAERR2: CMP     R0,#HIADRS
1151  004474  001005                  BNE     JAERR3
1152  004476  012737  177777  032312  MOV     #-1,HIAFLG
1153  004504  104056          1$:     ERROR   56              ;CACHE REGISTER RESPONSE TEST FAILED
1154  004506  000441                  BR      JAERR9
1155
1156  004510  020027  177744          JAERR3: CMP     R0,#MEMERR
1157  004514  001005                  BNE     JAERR4
1158  004516  012737  177777  032314  MOV     #-1,MMRFLG
1159  004524  104057          1$:     ERROR   57              ;CACHE REGISTER RESPONSE TEST FAILED
1160  004526  000431                  BR      JAERR9
1161
1162  004530  020027  177746          JAERR4: CMP     R0,#CONTRL
1163  004534  001005                  BNE     JAERR5
1164  004536  012737  177777  032316  MOV     #-1,CONFLG
1165  004544  104060          1$:     ERROR   60              ;CACHE REGISTER RESPONSE TEST FAILED
1166  004546  000421                  BR      JAERR9
1167
1168  004550  020027  177750          JAERR5: CMP     R0,#MAINT
1169  004554  001005                  BNE     JAERR6
1170  004556  012737  177777  032320  MOV     #-1,MANFLG
1171  004564  104061          1$:     ERROR   61              ;CACHE REGISTER RESPONSE TEST FAILED
1172  004566  000411                  BR      JAERR9
1173
1174  004570  020027  177752          JAERR6: CMP     R0,#HITMIS
1175  004574  001005                  BNE     JAERR7
1176  004576  012737  177777  032322  MOV     #-1,HIMFLG
1177  004604  104062          1$:     ERROR   62              ;CACHE REGISTER RESPONSE TEST FAILED
1178  004606  000401                  BR      JAERR9
1179
1180  004610  000000          JAERR7: HALT                    ;???
```

```
1181
1182  004612  005037  177766          JAERR9: CLR     @#CPUERR
1183  004616  000137  004352                  JMP     JA2
1184
1185  004622  005037  177766          JADONE: CLR     @#CPUERR        ;DONE!
1186
1187                                  ;;*********************************************************************
1188                                  ;*TEST 2         CACHE REGISTERS DATA PATH, READ ZEROES TEST
1189                                  ;*
1190                                  ;*THIS TEST CHECKS THE ABILITY OF THE CACHE REGISTER
1191                                  ;*DATA PATHS TO PASS 0'S BY FIRST WRITING THEN READING
1192                                  ;*0'S AT THE CONTROL AND MAINTENANCE REGISTERS.
1193                                  ;*
1194                                  ;;*********************************************************************
1195  004626  000004                  TST2:   SCOPE
1196          000002                  JB=$TN-1
1197                                                          ;SET THE SKAD REGISTER
1198  004630  012737  004770  032100          MOV     #TST3,SKAD      ;IN CASE THE TEST ABORTS.
1199
1200  004636  113737  001102  001224          MOVB    $TSTNM,$TMP0
1201  004644  012737  031754  000114          MOV     #SPUR,@#CACHVEC
1202  004652  005001                          CLR     R1              ;;INITIALIZE
1203
1204  004654  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1205  004656  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
1206  004660  012737  004666  001110          MOV     #JB1,$LPERR
1207  004666  005037  177746          JB1:    CLR     @#CONTRL        ;WRITE ZEROES
1208  004672  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
1209  004674  013700  177746          1$:     MOV     @#CONTRL,R0     ;READ,ZEROES
1210  004700  005700                          TST     R0
1211  004702  001432                          BEQ     JBDONE
1212  004704  005201                          INC     R1              ;;ON A PDP 11/ 74 WAIT
1213  004706  001372                          BNE     1$              ;;FOR THE VCIP BIT IN CACHE CONT.
1214                                                                  ;;REG TO CLEAR, IN CASE A FLUSH
1215                                                                  ;;WAS INITIATED BY CLEARING VSIU BIT
1216                                                                  ;;IN CACHE CONT. REG  (ABOVE)
1217  004710  005037  177750          JB2:    CLR     @#MAINT
1218  004714  013701  177750                  MOV     @#MAINT,R1
1219  004720  005701                          TST     R1
1220  004722  001414                          BEQ     JBERR2
1221
1222  004724                          JBERR1:                         ;BOTH READ ZEROES FAILED.
1223  004724  010037  001230                  MOV     R0,$TMP2
1224  004730  010137  001232                  MOV     R1,$TMP3
1225  004734  104063                  1$:     ERROR   63
1226  004736  012737  177777  032316          MOV     #-1,CONFLG      ;SIGNAL BAD REGISTERS
1227  004744  012737  177777  032320          MOV     #-1,MANFLG
1228  004752  000406                          BR      JBDONE
1229
1230  004754                          JBERR2:                         ;ONLY THE READ OF THE
1231  004754  010037  001230                  MOV     R0,$TMP2        ;CONTROL REGISTER FAILED.
1232  004760  104064                  1$:     ERROR   64
1233  004762  012737  177777  032316          MOV     #-1,CONFLG
1234
1235  004770                          JBDONE:                         ;DONE!!!
1236
```

```
                                ;;*****************************************************************
                                ;*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.
                                ;*
                                ;;*****************************************************************
004770  000004                  TST3:   SCOPE
004772  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000003                  JC=$TN-1
                                                                ;SET THE SKAD REGISTER
005000  012737  005132  032100          MOV     #TST4,SKAD      ;IN CASE THE TEST ABORTS.

005006  113737  001102  001224          MOVB    $TSTNM,$TMP0

005014  104426                          SKPBAD                  ;IF THE ERROR ADDRESS REG IS BAD SKIP THIS TEST.
005016  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
005020  012737  177777  177744          MOV     #-1,@#MEMERR    ;MAKE SURE THE ERROR REGISTERS ARE UNLOCKED
005026  012737  005034  001110          MOV     #JC1,$LPERR

005034  000240          JC1:    NOP                             ;FOR SCOPING WITH AN OSCILLOSCOPE!
005036  013700  177740                  MOV     @#LOADRS,R0
005042  013701  177742                  MOV     @#HIADRS,R1     ;READ THE REGISTERS.
005046  022700  177740                  CMP     #177740,R0
005052  001003                          BNE     JCERR1
005054  022701  000003          JC2:    CMP     #3,R1
005060  001424                          BEQ     JCDONE

005062  012737  005100  001226  JCERR1: MOV     #1$,$TMP1       ;BAD DATA WAS READ FROM THEM!!
005070  010037  001230                  MOV     R0,$TMP2
005074  010137  001232                  MOV     R1,$TMP3
005100  104065                  1$:     ERROR   65
005102  022700  000003                  CMP     #3,R0
005106  001403                          BEQ     2$
005110  012737  177777  032310          MOV     #-1,LOAFLG
005116  022700  177740          2$:     CMP     #177740,R0
005122  001403                          BEQ     JCDONE
005124  012737  177777  032312          MOV     #-1,HIAFLG

005132                          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.
```

```
                                        ;*
                                        ;;*****************************************************************
                                        TST4:   SCOPE
005132  000004
005134  012737  000004  001274          MOV     #4,$TIMES       ;;DO 4 ITERATIONS

        000004                          JD=$TN-1
                                                                ;SET THE SKAD REGISTER
005142  012737  005332  032100          MOV     #TST5,SKAD      ;IN CASE THE TEST ABORTS.

005150  113737  001102  001224          MOVB    $TSTNM,$TMP0

005156  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.

                                        ;;*****************************************************************
                                        ;*TEST 4         CATCHE CONTROL REGISTER PATTERN TEST
                                        ;*THIS TEST RUNS A COUNT PATTERN THROUGH THE LOWER 6 BITS OF THE CATCHE 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          ).
                                        ;;*****************************************************************
005160  012700  177746                  MOV     #CONTRL,R0      ;ADDRESS OF CONTRL TO R0
005164  005010                          CLR     (R0)            ;CLEAR CLR
005166  012702  000077                  MOV     #77,R2          ;INITIALIZE TEST PATTERN
005172  010210                  SBT1:   MOV     R2,(R0)         ;WRITE IT
005174  011001                          MOV     (R0),R1         ;READ IT BACK
005176  042701  177700                  BIC     #177700,R1      ;IGNORE <15:6>
005202  020201                          CMP     R2,R1           ;ARE THEY THE SAME?
005204  001040                          BNE     JDERR1          ;NO
005206  077207                  SBT1.2: SOB     R2,SBT1         ;YES, ITERATE
005210  005010                          CLR     (R0)            ;DONE WITH SUBTEST
005212  105737  001311                  TSTB    KB11EM          ;IS THIS A KB11-EM PROCESSOR?
005216  001003                          BNE     ST2             ;BR IF YES
005220  105737  001312                  TSTB    KB11CM          ;IS THIS A MODIFIED PROCESSOR (KB11CM)?
005224  001442                          BEQ     JDDONE          ;NO, GO TO END OF TEST.
005226  012702  001000          ST2:    MOV     #BIT9,R2        ;MARCH A BIT ACROSS THE REMAINING FIELDS
005232  010210                          MOV     R2,(R0)         ;WRITE
005234  011001                          MOV     (R0),R1         ;READ BACK
005236  001423                          BEQ     JDERR1          ;ERROR
005240  052737  000001  177750          BIS     #BIT0,@#MAINT   ;ALLOW THE DMMA BIT (CCR<11>) TO BE SET
005246  072227  000002                  ASH     #2,R2           ;SHIFT LEFT TWO
005252  010210                          MOV     R2,(R0)         ;WRITE DMMA
005254  011001                          MOV     (R0),R1         ;READ BACK
005256  001413                          BEQ     JDERR1          ;BAD.
005260  072227  000002                  ASH     #2,R2           ;SET UP TO TEST...
005264  010210                          MOV     R2,(R0)         ;VSIU
005266  011001                          MOV     (R0),R1
005270  001406                          BEQ     JDERR1
005272  006302                          ASL     R2              ;NOW TEST...
005274  010210                          MOV     R2,(R0)         ;IVSS
005276  011001                          MOV     (R0),R1
005300  001402                          BEQ     JDERR1          ;ERROR
005302  005010                          CLR     (R0)            ;DONE WITH TEST
005304  000412                          BR      JDDONE
```

```
1349  005306  010237  001230          JDERR1: MOV     R2,$TMP2        ;REPORT THE ERROR
1350  005312  010137  001232                  MOV     R1,$TMP3
1351  005316  010237  001234                  MOV     R2,$TMP4
1352  005322  104066                          ERROR   66
1353  005324  012737  177777  032316          MOV     #-1,CONFLG
1354  005332                          JDDONE:
1355
1356
1357                                    ;;*******************************************************************
1358                                    ;*TEST 5         CACHE HIT/MISS AND CONTROL REGISTER SIMPLE MISSES TEST
1359                                    ;*
1360                                    ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
1361                                    ;*CONTRL REGISTER'S ABILITY TO FORCE MISSES. ZEROES ARE
1362                                    ;*FLOATED THROUGH THE HIT/MISS REGISTER.
1363                                    ;*
1364                                    ;;*******************************************************************
1365  005332  000004          TST5:   SCOPE
1366  005334  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1367           000005          KB=$TN-1
1368                                                                  ;SET THE SKAD REGISTER
1369  005342  012737  005664  032100          MOV     #TST6,SKAD      ;IN CASE THE TEST ABORTS.
1370
1371  005350  113737  001102  001224          MOVB    $TSTNM,$TMP0
1372
1373
1374  005356  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1375  005360  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1376  005362  005037  005554                  CLR     KBFLG
1377  005366  012737  000014  177746  KB1:    MOV     #M0M1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
1378  005374  012737  005366  001110          MOV     #KB1,$LPERR
1379
1380  005402  012700  005412                  MOV     #KB2,R0
1381  005406  012701  000020                  MOV     #20,R1
1382  005412  005720                  KB2:    TST     (R0)+
1383  005414  077102                          SOB     R1,KB2
1384  005416  000240                          NOP                     ;GET SIX FORCED MISSES.
1385  005420  000240                          NOP
1386  005422  000240                          NOP
1387  005424  000240                          NOP
1388  005426  013702  177752                  MOV     @#HITMIS,R2     ;SHOULD HAVE REGISTERED
1389  005432  001051                          BNE     KBERR1          ;SIX MISSES.
1390
1391  005434  012737  005434  001110  KB3:    MOV     #KB3,$LPERR
1392  005442  012737  000054  177746          MOV     #S1M0M1,@#CONTRL        ;SELECT GROUP ONE, MISS GROUP
1393  005450  012700  005460                  MOV     #KB4,R0         ;ZERO AND GROUP ONE.
1394  005454  012701  000020                  MOV     #20,R1
1395  005460  005720                  KB4:    TST     (R0)+
1396  005462  077102                          SOB     R1,KB4
1397  005464  000240                          NOP
1398  005466  000240                          NOP
1399  005470  000240                          NOP
1400  005472  000240                          NOP
1401  005474  013702  177752                  MOV     @#HITMIS,R2     ;SHOULD HAVE SIX MISSES.
1402  005500  001035                          BNE     KBERR2
1403
1404  005502  012737  005502  001110  KB5:    MOV     #KB5,$LPERR
```

```
1405  005510  012737  000034  177746          MOV     #SOMOM1,@#CONTRL         ;SELECT GROUP 0, MISS GROUP 0
1406  005516  012700  005526                  MOV     #KB6,R0          ;AND GROUP 1.
1407  005522  012701  000020                  MOV     #20,R1
1408  005526  005720                  KB6:    TST     (R0)+
1409  005530  077102                          SOB     R1,KB6
1410  005532  000240                          NOP
1411  005534  000240                          NOP
1412  005536  000240                          NOP
1413  005540  000240                          NOP
1414  005542  013702  177752                  MOV     @#HITMIS,R2      ;SHOULD HAVE SIX MISSES.
1415  005546  001021                          BNE     KBERR3
1416  005550  000137  005626                  JMP     KBDONE
1417
1418
1419  005554  000000                  KBFLG:  .WORD   0                ;ERROR FLAG.
1420
1421  005556                          KBERR1:                          ;GOT HITS WHILE FORCING
1422  005556  010237  001230                  MOV     R2,$TMP2         ;MISSES TO BOTH GROUPS.
1423  005562  104072                  1$:     ERROR   72
1424  005564  052737  000001  005554          BIS     #BIT0,KBFLG
1425  005572  000720                          BR      KB3
1426  005574                          KBERR2:                          ;GO HITS WHILE FORCING
1427  005574  010237  001230                  MOV     R2,$TMP2         ;MISSES TO BOTH GROUPS
1428  005600  104073                  1$:     ERROR   73               ;AND SELECTING GROUP 1
1429  005602  052737  000002  005554          BIS     #BIT1,KBFLG
1430  005610  000734                          BR      KB5
1431  005612                          KBERR3:                          ;GO HITS WHILE FORCING
1432  005612  010237  001230                  MOV     R2,$TMP2         ;MISSES TO BOTH GROUPS
1433  005616  104074                  1$:     ERROR   74               ;AND SELECTING GROUP 0.
1434  005620  052737  000004  005554          BIS     #BIT2,KBFLG
1435
1436  005626  005037  177746          KBDONE: CLR     @#CONTRL
1437  005632  022737  000007  005554          CMP     #7,KBFLG         ;IF THE TEST DETECTED
1438  005640  001003                          BNE     KBD2             ;HITS FOR ALL OF THE
1439  005642  012737  177777  032336          MOV     #-1,HIMFL2       ;THREE CONDITION USED IN
1440                                                                   ;THE CONTROL REGISTER
1441                                                                   ;SIGNAL A BAD HIT/MISS
1442                                                                   ;REGISTER.
1443  005650  005737  005554          KBD2:   TST     KBFLG            ;IF LESS THEN THREE (BUT
1444  005654  001403                          BEQ     KBD3             ;MORE THAN ZERO) CONTRL
1445  005656  012737  177777  032332          MOV     #-1,CONFL2       ;PATTERNS FAILED SIGNAL
1446                                                                   ;A BAD CONTROL REGISTER.
1447  005664                          KBD3:                            ;DONE!
1448
1449                                  ;;*********************************************************************
1450                                  ;*TEST 6         CACHE HIT/MISS AND CONTROL REGISTER SIMPLE HIT TEST
1451                                  ;*
1452                                  ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
1453                                  ;*THE FORCE MISS BITS OF THE CONTROL REGISTER.
1454                                  ;*WHAT IS DONE IS TO SEE IF ANY HITS AT ALL ARE
1455                                  ;*POSSIBLE WITH THE CONTROL REGISTER CLEARED.  THEN THE
1456                                  ;*SAME IS DONE WITH EACH GROUP DISABLE ONE AT A TIME.
1457                                  ;*BY DISABLED IS MEANT THAT THE FORCE MISS BIT IS SET
1458                                  ;*IN THE CONTROL REGISTER FOR THE DISABLED GROUP AND THE
1459                                  ;*FORCE SELECT BIT IS SET FOR THE OTHER GROUP.
1460                                  ;*
```

```
1461                                        ;;*****************************************************************
1462  005664  000004                        TST6:   SCOPE
1463  005666  012737  000040  001274                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1464           000006                  KA=$TN-1
1465                                                                        ;SET THE SKAD REGISTER
1466  005674  012737  006234  032100                MOV     #TST7,SKAD      ;IN CASE THE TEST ABORTS.
1467
1468  005702  113737  001102  001224                MOVB    $TSTNM,$TMP0
1469
1470
1471  005710  104432                              SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1472  005712  104436                              SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1473  005714  005037  006120                      CLR     KAFLG
1474  005720  005037  177746                KA1:  CLR     @#CONTRL        ;BOTH GROUPS ENABLED.
1475  005724  012737  005720  001110              MOV     #KA1,$LPERR
1476  005732  012700  005742                      MOV     #KA2,R0
1477  005736  012701  000020                      MOV     #20,R1
1478
1479  005742  005720                KA2:  TST     (R0)+           ;SET UP HITS IN BOTH
1480  005744  077102                      SOB     R1,KA2          ;GROUPS
1481  005746  000240                      NOP
1482  005750  000240                      NOP
1483  005752  000240                      NOP
1484  005754  000240                      NOP
1485  005756  013702  177752              MOV     @#HITMIS,R2     ;SHOULD HAVE ALL HITS.
1486  005762  022702  000077              CMP     #77,R2
1487  005766  001055                      BNE     KAERR1
1488
1489  005770  012737  005770  001110  KA3:  MOV     #KA3,$LPERR
1490  005776  012737  000044  177746        MOV     #S1M0,@#CONTRL  ;DISABLE GROUP ZERO.
1491  006004  012700  006014              MOV     #KA4,R0
1492  006010  012701  000020              MOV     #20,R1
1493  006014  005720                KA4:  TST     (R0)+           ;SET UP HITS IN GROUP 1
1494  006016  077102                      SOB     R1,KA4
1495  006020  000240                      NOP
1496  006022  000240                      NOP
1497  006024  000240                      NOP
1498  006026  000240                      NOP
1499  006030  013702  177752              MOV     @#HITMIS,R2     ;SHOULD HAVE ALL HITS.
1500  006034  022702  000077              CMP     #77,R2
1501  006040  001037                      BNE     KAERR2
1502  006042  012737  006042  001110  KA5:  MOV     #KA5,$LPERR
1503  006050  012737  000030  177746        MOV     #S0M1,@#CONTRL  ;DISABLE GROUP ONE.
1504  006056  012700  006066              MOV     #KA6,R0
1505  006062  012701  000020              MOV     #20,R1
1506  006066  005720                KA6:  TST     (R0)+           ;SET UP HITS IN GROUP ZERO.
1507  006070  077102                      SOB     R1,KA6
1508  006072  000240                      NOP
1509  006074  000240                      NOP
1510  006076  000240                      NOP
1511  006100  000240                      NOP
1512  006102  013702  177752              MOV     @#HITMIS,R2     ;SHOULD HAVE SIX HITS.
1513  006106  022702  000077              CMP     #77,R2
1514  006112  001021                      BNE     KAERR3
1515  006114  000137  006172              JMP     KADONE
1516
```

```
1517  006120  000000                    KAFLG:  .WORD   0               ;ERROR FLAG.
1518
1519  006122                            KAERR1:                         ;FAILED TO GET HITS
1520  006122  010237  001230                    MOV     R2,$TMP2        ;WITH THE CONTROL
1521  006126  104067                    1$:     ERROR   67              ;REGISTER CLEAR!
1522  006130  052737  000001  006120            BIS     #BIT0,KAFLG
1523  006136  000714                            BR      KA3
1524  006140                            KAERR2:                         ;FAILED TO GET HITS
1525  006140  010237  001230                    MOV     R2,$TMP2        ;WITH THE CONTROL REGISTER
1526  006144  104070                    1$:     ERROR   70              ;SET TO FORCE SELECT GROUP
1527  006146  052737  000002  006120            BIS     #BIT1,KAFLG     ;ONE FORCE MISS GROUP ZERO.
1528  006154  000732                            BR      KA5
1529  006156                            KAERR3:                         ;FAILED TO GET HITS
1530  006156  010237  001230                    MOV     R2,$TMP2        ;WITH THE CONTROL REGISER
1531  006162  104071                    1$:     ERROR   71              ;SET TO FORCE SELECT GROUP
1532  006164  052737  000004  006120            BIS     #BIT2,KAFLG     ;ZERO AND FORCE MISS GROUP ONE.
1533  006172  005037  177746            KADONE: CLR     @#CONTRL
1534  006176  022737  000007  006120            CMP     #7,KAFLG        ;IF THE TEST FAILED FOR ALL
1535  006204  001004                            BNE     KAD2            ;THREE CONDITIONS OF THE
1536  006206  012737  177777  032322            MOV     #-1,HIMFLG      ;CONTROL REGISTER SIGNAL
1537  006214  000407                            BR      KAD3            ;A BAD HIT/MISS REGISTER.
1538
1539  006216  032737  000006  006120  KAD2:     BIT     #6,KAFLG        ;IF THE TEST FAILED ONLY WHEN
1540  006224  001403                            BEQ     KAD3            ;THE CONTROL REGISTER WAS SET
1541  006226  012737  177777  032332            MOV     #-1,CONFL2      ;SIGNAL A BAD CONTROL REGISTER.
1542  006234                            KAD3:                           ;DONE!!
1543
1544
1545                                    ;;**************************************************************************
1546                                    ;*TEST 7        CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 0 TEST
1547                                    ;*
1548                                    ;*THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS
1549                                    ;*OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS
1550                                    ;*MADE A HIT IN GROUP ONE; THEN ANOTHER ADDRESS, WHOSE
1551                                    ;*HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS
1552                                    ;*IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING
1553                                    ;*SELECTION OF GROUP ZERO; THEN SEE IF THE FIRST ADDRESS
1554                                    ;*IS STILL A HIT IN GROUP ONE; FINALLY TURN ON THE FORCE
1555                                    ;*MISS GROUP ZERO BIT AND SEE IF THE SECOND ADDRESS'
1556                                    ;*HIT IN GROUP ZERO CAN BE FORCED TO A MISS.
1557                                    ;*
1558                                    ;;**************************************************************************
1559  006234  000004                    TST7:   SCOPE
1560  006236  012737  000040  001274            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1561          000007                    KD=$TN-1
1562                                                                    ;SET THE SKAD REGISTER
1563  006244  012737  006564  032100            MOV     #TST10,SKAD     ;IN CASE THE TEST ABORTS.
1564
1565  006252  113737  001102  001224            MOVB    $TSTNM,$TMP0
1566  006260  012737  031754  000114            MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS.
1567
1568  006266  104432                            SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1569  006270  104436                            SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1570
1571  006272  012700  006562            K1D:    MOV     #KTMP2D,R0      ;DETERMINE THE TEST LOCATIONS.
1572  006276  042700  176003                    BIC     #176003,R0
```

```
1573  006302  010001                        MOV     R0,R1
1574  006304  062701  140000                ADD     #TESTR1,R1
1575  006310  010137  001244                MOV     R1,$TMP10
1576  006314  005037  001246                CLR     $TMP11
1577  006320  010002                        MOV     R0,R2
1578  006322  062702  142000                ADD     #TESTR2,R2
1579  006326  010237  001250                MOV     R2,$TMP12
1580  006332  005037  001252                CLR     $TMP13
1581
1582  006336  012737  000044  177746  K2D:  MOV     #S1M0,@#CONTRL  ;MAKE (R1) A HIT IN
1583  006344  005711                        TST     (R1)            ;GROUP GRM.
1584  006346  005711                        TST     (R1)
1585  006350  032737  000010  177752        BIT     #10,@#HITMIS
1586  006356  001007                        BNE     K3D
1587
1588                                                                ;REPORT ERROR, UNABLE
1589  006360  012737  000001  001230        MOV     #1,$TMP2        ;GET A HIT IN GROUP GRM.
1590  006366  012737  000044  001232        MOV     #S1M0,$TMP3
1591  006374  104075                1$:     ERROR   75
1592
1593  006376  012703  000030        K3D:    MOV     #S0M1,R3
1594  006402  042703  000017                BIC     #17,R3
1595  006406  010337  177746                MOV     R3,@#CONTRL     ;FORCE SELECT GROUP GRS.
1596  006412  005712                        TST     (R2)            ;MAKE (R2) A HIT IN GROUP
1597  006414  005712                        TST     (R2)            ;GRS.
1598  006416  032737  000010  177752        BIT     #10,@#HITMIS
1599  006424  001006                        BNE     K4D
1600                                                                ;IF NOT, ERROR UNABLE TO
1601                                                                ;GET A HIT IN GROUP 0
1602  006426  010337  001232                MOV     R3,$TMP3
1603  006432  104076                1$:     ERROR   76
1604  006434  012737  177777  032332        MOV     #-1,CONFL2
1605
1606  006442  005037  177746        K4D:    CLR     @#CONTRL        ;NOW MAKE SURE (R1) IS
1607  006446  000240                        NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
1608  006450  005711                        TST     (R1)            ;STILL A HIT IN GROUP
1609  006452  032737  000010  177752        BIT     #10,@#HITMIS    ;1, THAT IS MAKE SURE
1610  006460  001010                        BNE     K5D             ;GROUP 1 WASN'T WRITTEN
1611                                                                ;WHILE FORCE SELECTING
1612                                                                ;GROUP GRS.
1613  006462  012737  000001  001230        MOV     #1,$TMP2
1614  006470  012737  000000  001232        MOV     #0,$TMP3
1615  006476  104077                1$:     ERROR   77
1616  006500  000424                        BR      K6D
1617  006502  012703  000044        K5D:    MOV     #S1M0,R3        ;NOW SEE IF YOU CAN
1618  006506  042703  000063                BIC     #63,R3          ;GET A MISS AT (R2)
1619  006512  010337  177746                MOV     R3,@#CONTRL     ;BY FORCING MISSES
1620  006516  005712                        TST     (R2)            ;TO GRS.
1621  006520  032737  000010  177752        BIT     #10,@#HITMIS
1622  006526  001411                        BEQ     K6D             ;SHOULD BE A MISS.
1623                                                                ;OTHERWISE ERROR!
1624  006530  012737  000000  001230        MOV     #0,$TMP2
1625  006536  010337  001232                MOV     R3,$TMP3
1626  006542  104117                1$:     ERROR   117
1627  006544  012737  177777  032332        MOV     #-1,CONFL2
1628
```

```
1629  006552  005037  177746          K6D:    CLR     @#CONTRL
1630  006556  000402                          BR      K7D
1631
1632  006560  000000                  KTMP1D:.WORD    0
1633  006562  000000                  KTMP2D:.WORD    0
1634
1635  006564                          K7D:                            ;DONE!
1636
1637
1638                                  ;;*****************************************************************
1639                                  ;*TEST 10       CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 1 TEST
1640                                  ;*
1641                                  ;*THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS
1642                                  ;*OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS
1643                                  ;*MADE A HIT IN GROUP ZERO; THEN ANOTHER ADDRESS, WHOSE
1644                                  ;*HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS
1645                                  ;*IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING
1646                                  ;*SELECTION OF GROUP ONE; THEN SEE IF THE FIRST ADDRESS
1647                                  ;*IS STILL A HIT IN GROUP ZERO; FINALLY TURN ON THE FORCE
1648                                  ;*MISS GROUP ONE BIT AND SEE IF THE SECOND ADDRESS'
1649                                  ;*HIT IN GROUP ONE CAN BE FORCED TO A MISS.
1650                                  ;*
1651                                  ;;*****************************************************************
1652  006564  000004                  TST10:  SCOPE
1653  006566  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1654           000010                  KE=$TN-1
1655                                                                  ;SET THE SKAD REGISTER
1656  006574  012737  007114  032100          MOV     #TST11,SKAD     ;IN CASE THE TEST ABORTS.
1657
1658  006602  113737  001102  001224          MOVB    $TSTNM,$TMP0
1659  006610  012737  031754  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS.
1660
1661  006616  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1662  006620  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1663
1664  006622  012700  007112          K1E:    MOV     #KTMP2E,R0      ;DETERMINE THE TEST LOCATIONS.
1665  006626  042700  176003                  BIC     #176003,R0
1666  006632  010001                          MOV     R0,R1
1667  006634  062701  140000                  ADD     #TESTR1,R1
1668  006640  010137  001244                  MOV     R1,$TMP10
1669  006644  005037  001246                  CLR     $TMP11
1670  006650  010002                          MOV     R0,R2
1671  006652  062702  142000                  ADD     #TESTR2,R2
1672  006656  010237  001250                  MOV     R2,$TMP12
1673  006662  005037  001252                  CLR     $TMP13
1674
1675  006666  012737  000030  177746  K2E:    MOV     #SOM1,@#CONTRL  ;MAKE (R1) A HIT IN
1676  006674  005711                          TST     (R1)            ;GROUP GRM.
1677  006676  005711                          TST     (R1)
1678  006700  032737  000010  177752          BIT     #10,@#HITMIS
1679  006706  001007                          BNE     K3E
1680
1681                                                                  ;REPORT ERROR. UNABLE
1682  006710  012737  000000  001230          MOV     #0,$TMP2        ;GET A HIT IN GROUP GRM.
1683  006716  012737  000030  001232          MOV     #SOM1,$TMP3
1684  006724  104075                  1$:     ERROR   75
```

```
1685
1686  006726  012703  000044          K3E:    MOV     #S1M0,R3
1687  006732  042703  000017                  BIC     #17,R3
1688  006736  010337  177746                  MOV     R3,@#CONTRL     ;FORCE SELECT GROUP GRS.
1689  006742  005712                          TST     (R2)            ;MAKE (R2) A HIT IN GROUP
1690  006744  005712                          TST     (R2)            ;GRS.
1691  006746  032737  000010  177752          BIT     #10,@#HITMIS
1692  006754  001006                          BNE     K4E
1693                                                                  ;IF NOT, ERROR UNABLE TO
1694                                                                  ;GET A HIT IN GROUP 1
1695  006756  010337  001232                  MOV     R3,$TMP3
1696  006762  104076                  1$:     ERROR   76
1697  006764  012737  177777  032332          MOV     #-1,CONFL2
1698
1699  006772  005037  177746          K4E:    CLR     @#CONTRL        ;NOW MAKE SURE (R1) IS
1700  006776  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
1701  007000  005711                          TST     (R1)            ;STILL A HIT IN GROUP
1702  007002  032737  000010  177752          BIT     #10,@#HITMIS    ;0, THAT IS MAKE SURE
1703  007010  001010                          BNE     K5E             ;GROUP 0 WASN'T WRITTEN
1704                                                                  ;WHILE FORCE SELECTING
1705                                                                  ;GROUP GRS.
1706  007012  012737  000000  001230          MOV     #0,$TMP2
1707  007020  012737  000001  001232          MOV     #1,$TMP3
1708  007026  104077                  1$:     ERROR   77
1709  007030  000424                          BR      K6E
1710  007032  012703  000030          K5E:    MOV     #S0M1,R3        ;NOW SEE IF YOU CAN
1711  007036  042703  000063                  BIC     #63,R3          ;GET A MISS AT (R2)
1712  007042  010337  177746                  MOV     R3,@#CONTRL     ;BY FORCING MISSES
1713  007046  005712                          TST     (R2)            ;TO GRS.
1714  007050  032737  000010  177752          BIT     #10,@#HITMIS
1715  007056  001411                          BEQ     K6E             ;SHOULD BE A MISS,
1716                                                                  ;OTHERWISE ERROR!
1717  007060  012737  000001  001230          MOV     #1,$TMP2
1718  007066  010337  001232                  MOV     R3,$TMP3
1719  007072  104117                  1$:     ERROR   117
1720  007074  012737  177777  032332          MOV     #-1,CONFL2
1721
1722  007102  005037  177746          K6E:    CLR     @#CONTRL
1723  007106  000402                          BR      K7E
1724
1725  007110  000000          KTMP1E:.WORD    0
1726  007112  000000          KTMP2E:.WORD    0
1727
1728  007114                  K7E:                                    ;DONE!
1729
1730
1731                          ;;*************************************************************************
1732                          ;*TEST 11       CACHE HIT/MISS REGISTER PATTERNS TEST
1733                          ;*
1734                          ;*THIS IS A TEST OF THE HIT/MISS REGISTER WHICH
1735                          ;*FLOATS DIFFERENT PATTERNS OF HITS AND MISSES
1736                          ;*THROUGH THAT REGISTER. THIS IS DONE FIRST WITH
1737                          ;*BOTH GROUPS ENABLE; THEN WITH GROUP ZERO DISABLED
1738                          ;*THAT IS FORCING SELECTION OF GROUP ONE AND FORCING
1739                          ;*MISSES TO GROUP ZERO; FINALLY WITH GROUP ONE
1740                          ;*DISABLED.
```

```
                                        ;*
                                        ;;*************************************************************************
007114  000004                  TST11:  SCOPE
007116  012737  000020  001274          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000011                  KC=$TN-1
                                                                ;SET THE SKAD REGISTER
007124  012737  007724  032100          MOV     #TST12,SKAD     ;IN CASE THE TEST ABORTS.

007132  113737  001102  001224          MOVB    $TSTNM,$TMPO
007140  012737  031754  000114          MOV     #SPUR,@#CACHVEC

007146  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
007150  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
007152  005037  007606                  CLR     KCCON           ;TEST THE BOTH GROUPS
007156  012737  000002  007610          MOV     #2,KCFLG1       ;ENABLED CONDITION FIRST.
007164  012737  007200  001110  KC0:    MOV     #KC1,$LPERR
007172  012737  007614  007612          MOV     #KCTBL,KCPTR    ;KCPTR IS A POINTER TO
                                                                ;THE TABLE OF 12-BIT PATTERNS
                                                                ;WHICH WILL BE FLOATED
                                                                ;THROUGH THE REGISTER.

007200  012701  140000          KC1:    MOV     #TESTR1,R1      ;MAKE THIS CODE MISSES
007204  012702  142000                  MOV     #TESTR2,R2      ;TO BOTH GROUPS!
007210  012700  001000                  MOV     #1000,R0
007214  012737  000030  177746  1$:     MOV     #S0M1,@#CONTRL
007222  005721                          TST     (R1)+
007224  012737  000044  177746          MOV     #S1M0,@#CONTRL
007232  005722                          TST     (R2)+
007234  077011                          SOB     R0,1$

007236  017702  000350                  MOV     @KCPTR,R2       ;GET THE HIT/MISS PATTERN
007242  012700  007322                  MOV     #KC3,R0         ;AND MAKE THE INSTRUCTIONS
007246  012701  000007                  MOV     #7,R1           ;BETWEEN KC3 AND KC9
007252  013737  007606  177746          MOV     KCCON,@#CONTRL  ;HITS AND MISSES SO THAT
007260  000403                          BR      KC2.5           ;WHEN THAT CODE IS EXECUTED
007262  006302                  KC2:    ASL     R2              ;THIS PATTERN WILL BE FLOATED
007264  103001                          BCC     KC2.5           ;THROUGH THE HIT/MISS REGISTER.
007266  005710                          TST     (R0)            ;MAKE (R0) A HIT!
007270  062700  000002          KC2.5:  ADD     #2,R0
007274  006302                          ASL     R2
007276  103001                          BCC     1$
007300  005710                          TST     (R0)            ;MAKE (R0) A HIT!
007302  062700  000006          1$:     ADD     #6,R0
007306  077113                          SOB     R1,KC2

007310  012705  177752                  MOV     #HITMIS,R5      ;NOW THAT THE HITS
007314  000402                          BR      KC3             ;AND MISSES HAVE BEEN
                                                                ;APPROPRIATELY ESTABLISHED
                                                                ;EXECUTE THE CODE AND
                                                                ;CAUSE THE PATTERN TO FLOAT
                                                                ;THROUGH THE HIT/MISS
                                                                ;REGISTER.

        007316                          LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        007314                          LOC=-4&LOC
```

```
1797            007320                          LOC=LOC+4
1798            007320                          .=LOC
1799
1800  007320    000000                          HALT
1801  007322    000240                  KC3:    NOP                     ;THE HALT'S HERE ARE NOT
1802  007324    000402                          BR      KC4             ;EXECUTED, THEY ARE FILLERS.
1803  007326    000000                          HALT                    ;THE ADDRESS OF THE HIT AND
1804  007330    000000                          HALT                    ;MISS REGISTER IS IN R5.
1805  007332    011500                  KC4:    MOV     (R5),R0         ;NOTE THAT THE HIT/MISS
1806  007334    000402                          BR      KC5             ;REGISTER IS READ EVERY
1807  007336    000000                          HALT                    ;TWO CYCLES AND SAVED IN
1808  007340    000000                          HALT                    ;A PROCESSOR GENERAL
1809  007342    011501                  KC5:    MOV     (R5),R1         ;PURPOSE REGISTER.
1810  007344    000402                          BR      KC6
1811  007346    000000                          HALT
1812  007350    000000                          HALT
1813  007352    011502                  KC6:    MOV     (R5),R2
1814  007354    000402                          BR      KC7
1815  007356    000000                          HALT
1816  007360    000000                          HALT
1817  007362    011503                  KC7:    MOV     (R5),R3
1818  007364    000402                          BR      KC8
1819  007366    000000                          HALT
1820  007370    000000                          HALT
1821  007372    011504                  KC8:    MOV     (R5),R4
1822  007374    000402                          BR      KC9
1823  007376    000000                          HALT
1824  007400    000000                          HALT
1825  007402    011505                  KC9:    MOV     (R5),R5         ;CAN SAVE PATTERN IN R5
1826                                                                    ;SINCE THE ADDRESS IS
1827                                                                    ;NO LONGER NEEDED.
1828  007404    042700    177774        KC10:   BIC     #177774,R0      ;GET THE PATTERNS READ
1829  007410    010037    007640                MOV     R0,KCR0         ;FROM THE HIT/MISS REGISTER
1830  007414    042701    017760                BIC     #17760,R1       ;INTO LOCATIONS KCR0
1831  007420    010137    007642                MOV     R1,KCR1         ;THROUGH KCR5 SO THE
1832  007424    010237    007644                MOV     R2,KCR2         ;GENERAL PURPOSE REGISTERS
1833  007430    010337    007646                MOV     R3,KCR3         ;CAN BE USED FOR OTHER
1834  007434    010437    007650                MOV     R4,KCR4         ;THINGS
1835  007440    010537    007652                MOV     R5,KCR5
1836
1837  007444    017701    000142        KC11:   MOV     @KCPTR,R1
1838  007450    005000                          CLR     R0
1839  007452    012702    000006                MOV     #6,R2           ;PUT THE EXPECTED VALUES
1840  007456    012703    007654                MOV     #KCE0,R3        ;IN KCE0 THROUGH KCE5!
1841  007462    073027    000002        KC12:   ASHC    #2,R0
1842  007466    042700    177700                BIC     #177700,R0
1843  007472    010023                          MOV     R0,(R3)+
1844  007474    077206                          SOB     R2,KC12
1845
1846  007476    012700    007640                MOV     #KCR0,R0
1847  007502    012701    007654                MOV     #KCE0,R1        ;MAKE SURE THE PATTERNS
1848  007506    012702    000006                MOV     #6,R2           ;WHICH WERE READ FROM
1849  007512    022021                  KC13:   CMP     (R0)+,(R1)+     ;THE HIT AND MISS REGISTER
1850  007514    001402                          BEQ     KC14            ;MATCH THE EXPECTED
1851  007516    000137    007670                JMP     KCERR           ;PATTERNS.
1852  007522    077205                  KC14:   SOB     R2,KC13
```

```
1853
1854  007524  062737  000002  007612  KC15:   ADD     #2,KCPTR                ;MOVE POINTER TO NEXT
1855  007532  023727  007612  007636          CMP     KCPTR,#KCTBLB   ;PATTERN AND IF ALL THE
1856  007540  001402                          BEQ     1$              ;PATTERNS HAVEN'T BEEN
1857  007542  000137  007200                  JMP     KC1             ;TESTED GO TO KC1 TO TEST
1858                                                                  ;THIS NEXT PATTERN.
1859  007546  005337  007610          1$:     DEC     KCFLG1          ;IF ALL THE PATERNS HAVE BEEN
1860  007552  100002                          BPL     KC16            ;TESTED WITH THAT GROUP CONFIGURATION
1861  007554  000137  007720                  JMP     KCDONE          ;SO GO TO THE NEXT CONFIGURATION.
1862                                                                  ;OR DONE!!
1863  007560  001405                  KC16:   BEQ     KC17
1864  007562  012737  000044  007606          MOV     #S1M0,KCCON     ;BOTH GROUPS ENABLED CONFIGURATION
1865  007570  000137  007164                  JMP     KC0             ;HAS BEEN TESTED SO NOW TEST GROUP
1866                                                                  ;ZERO DISABLED CONFIGURATION.
1867  007574  012737  000030  007606  KC17:   MOV     #S0M1,KCCON     ;BOTH GROUPS ENABLED AND GROUP ZERO
1868                                                                  ;DISABLED CONFIGURATIONS HAVE BOTH
1869                                                                  ;BEEN TESTED SO FINALLY TEST THE
1870  007602  000137  007164                  JMP     KC0             ;GROUP ONE DISABLED CONFIGURATION.
1871
1872
1873
1874  007606  000000                  KCCON:  .WORD   0               ;PATTERN BEING USED IN THE CONTROL REGISTER
1875
1876  007610  000000                  KCFLG1: .WORD   0               ;FLAG USED TO DETERMINE THE CONFIGURATION
1877                                                                  ;BEING TESTED.
1878  007612  000000                  KCPTR:  .WORD   0               ;POINTER USED TO POINT TO THE PATTERN
1879                                                                  ;BEING TESTED IN KCTBL.
1880
1881  007614  000000                  KCTBL:  .WORD   0               ;PATTERNS WHICH ARE
1882  007616  002000                          .WORD   002000          ;FLOATED THROUGH THE HIT/MISS
1883  007620  177760                          .WORD   177760          ;REGISTER. ONLY THE UPPER
1884  007622  175760                          .WORD   175760          ;12 BITS HAVE ANY SIGNIFICANCE!!
1885  007624  125240                          .WORD   125240
1886  007626  146300                          .WORD   146300
1887  007630  161600                          .WORD   161600
1888  007632  100020                          .WORD   100020
1889  007634  077740                          .WORD   077740
1890  007636  000000                  KCTBLB: .WORD   0
1891
1892  007640  000000                  KCR0:   .WORD   0               ;STORAGE FOR THE PATTERNS READ
1893  007642  000000                  KCR1:   .WORD   0               ;OUT OF THE HUIT/MISS REGISTER.
1894  007644  000000                  KCR2:   .WORD   0
1895  007646  000000                  KCR3:   .WORD   0
1896  007650  000000                  KCR4:   .WORD   0
1897  007652  000000                  KCR5:   .WORD   0
1898
1899  007654  000000                  KCE0:   .WORD   0               ;EXPECTED VALUES FOR THE PATTERNS
1900  007656  000000                  KCE1:   .WORD   0               ;READ FROM THE HIT/MISS REGISTER.
1901  007660  000000                  KCE2:   .WORD   0
1902  007662  000000                  KCE3:   .WORD   0
1903  007664  000000                  KCE4:   .WORD   0
1904  007666  000000                  KCE5:   .WORD   0
1905
1906  007670                  KCERR:                                  ;REPORT THE PATTERN READ FROM THE
1907  007670  013737  007606  001230          MOV     KCCON,$TMP2     ;HIT/MISS REGISTER WAS NOT THE EXPECTED
1908  007676  104120                  1$:     ERROR   120             ;VALUE.
```

```
1909  007700  012737  177777  032332          MOV     #-1,CONFL2
1910  007706  012737  177777  032336          MOV     #-1,HIMFL2
1911  007714  000137  007524                  JMP     KC15
1912
1913  007720  005037  177746          KCDONE: CLR     @#CONTRL        ;DONE!!
1914
1915                                  ;;*****************************************************************
1916                                  ;*TEST 12       CACHE CONTROL AND HIT/MISS REGISTERS EVALUATION ROUTINE
1917                                  ;*
1918                                  ;*THIS IS NOT A TEST. THIS ROUTINE IS USED TO LOOK AT THE RESULTS
1919                                  ;*OF TST5 THROUGH TST10, WHICH TESTED THE HIT/MISS REGISTER
1920                                  ;*AND THE CONTROL REGISTER. THOSE TESTS HAVE SIGNALLED A BAD
1921                                  ;*REGISTER USING THE FLAGS, CONFL2 AND HIMFL2, REPRESENTING THE
1922                                  ;*CONTROL AND HIT/MISS REGISTERS RESPECTIVELY. IF ONE OF THESE
1923                                  ;*REGISTERS WAS FOUND TO BE BAD THE FLAG SHOULD BE A -1. WHILE A
1924                                  ;*ZERO FLAG INDICATES THAT THOSE TESTS FOUND THAT REGISTER
1925                                  ;*FUNCTIONAL. THIS ROUTINE LOOKS AT THE FLAGS, CONFL2 AND HIMFL2,
1926                                  ;*WHICH ARE CONSIDERED TO BE LOCAL AND TRANSFERS THE INDICATORS
1927                                  ;*THEY CONTAIN TO THE GLOBAL FLAGS, CONFLG AND HIMFLG. THESE GLOBAL
1928                                  ;*FLAGS ARE USED TO DESIGNATE TO THE REST OF THE PROGRAM THE FUNCTIONALITY
1929                                  ;*OR DISFUNCTIONALITY OF THOSE REGISTERS.
1930                                  ;*
1931                                  ;;*****************************************************************
1932  007724  000004                  TST12:  SCOPE
1933          000012                  KY=$TN-1
1934  007726  005737  032332                  TST     CONFL2
1935  007732  001403                          BEQ     KY1
1936  007734  012737  177777  032316          MOV     #-1,CONFLG
1937  007742  005737  032336          KY1:    TST     HIMFL2
1938  007746  001403                          BEQ     KY2
1939  007750  012737  177777  032322          MOV     #-1,HIMFLG
1940  007756                          KY2:                            ;DONE
1941
1942                                  ;;*****************************************************************
1943                                  ;*TEST 13       CACHE CONTROL LOGIC, 'RANDOM' FLIP FLOP TEST
1944                                  ;*
1945                                  ;*THIS IS A TEST OF THE 'RANDOM' CONTROL SIGNAL.
1946                                  ;*A TEST IS MADE TO INSURE THAT THE 'RANDOM' FLIP-FLOP IS NOT STUCK
1947                                  ;*AND IS TOGGLED ONCE FOR EVERY 'BUST' CYCLE INITIATED BY
1948                                  ;*THE PROCESSOR. 'BUST' IS BUS START, A SIGNAL PRODUCED BY
1949                                  ;*THE PROCESSOR WHENEVER IT THINKS IT IS ABOUT TO DO A MEMORY CYCLE.
1950                                  ;*THE RANDOM FLIP FLOP IS USED IN THE CACHE TO DETERMINE WHICH
1951                                  ;*GROUP TO WRITE IN THE EVENT OF A READ MISS CYCLE. IF THIS FLIP FLOP IS
1952                                  ;*SET THEN GROUP ZERO IS WRITTEN; IF CLEAR THEN GROUP ONE IS WRITTEN.
1953                                  ;*
1954                                  ;;*****************************************************************
1955  007756  000004                  TST13:  SCOPE
1956  007760  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1957          000013                  KF=$TN-1
1958                                                                  ;SET THE SKAD REGISTER
1959  007766  012737  010212  032100          MOV     #TST14,SKAD     ;IN CASE THE TEST ABORTS.
1960
1961  007774  113737  001102  001224          MOVB    $TSTNM,$TMP0
1962  010002  012737  031754  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
1963
1964  010010  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
```

```
1965  010012  104436                        SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1966  010014  012700  010210          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  010020  042700  176003                BIC     #176003,R0
1971  010024  010001                        MOV     R0,R1
1972  010026  062701  140000                ADD     #TESTR1,R1
1973  010032  010002                        MOV     R0,R2
1974  010034  062702  142000                ADD     #TESTR2,R2
1975
1976  010040  012737  000044  177746        MOV     #S1M0,@#CONTRL  ;MAKE THOSE TWO TEST LOCATIONS
1977  010046  005710                        TST     (R0)            ;(R1) AND (R2) MISSES IN BOTH
1978                                                              ;GROUPS BY MAKING (R0) A HIT
1979                                                              ;IN BOTH GROUPS.
1980
1981  010050  005710                        TST     (R0)
1982
1983                                                              ;SEE IF REFERENCE ADDRESS
1984  010052  032737  000010  177752        BIT     #10,@#HITMIS    ;IS A HIT.
1985  010060  001006                        BNE     KF2
1986                                                              ;IF NOT ERROR.
1987  010062  010037  001230                MOV     R0,$TMP2
1988  010066  012737  000001  001226        MOV     #1,$TMP1
1989  010074  104001                        ERROR   1
1990
1991
1992
1993
1994  010076  012737  000030  177746  KF2:  MOV     #S0M1,@#CONTRL
1995  010104  005710                        TST     (R0)
1996
1997  010106  005710                        TST     (R0)
1998
1999                                                              ;SEE IF REFERENCE ADDRESS
2000  010110  032737  000010  177752        BIT     #10,@#HITMIS    ;IS A HIT.
2001  010116  001006                        BNE     KF3
2002                                                              ;IF NOT ERROR!
2003  010120  010037  001230                MOV     R0,$TMP2
2004  010124  012737  000000  001226        MOV     #0,$TMP1
2005  010132  104001                        ERROR   1
2006
2007
2008
2009
2010  010134  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
```

```
                                                        ;WOULD BE MUTUALLY EXCLUSIVE,
                                                        ;THAT IS BOTH THESE ADDRESSES
                                                        ;CAN'T BE HITS IN THE SAME GROUP!

010140  000240                         NOP             ;FOR SCOPING WITH AN OSCILLOSCOPE!
010142  021112                         CMP     (R1),(R2)       ;HERE BOTH THE OPERAND FETCHES
                                                        ;SHOULD BE MISSES.
010144  021112                         CMP     (R1),(R2)       ;HERE BOTH THE OPERAND FETCHES
                                                        ;SHOULD BE HITS!
010146  013705  177752                 MOV     @#HITMIS,R5
010152  005105                         COM     R5
010154  032705  000014                 BIT     #14,R5          ;BOTH HITS ELSE ERROR.
010160  001411                         BEQ     KF4

010162  010137  001230                 MOV     R1,$TMP2        ;REPORT THE ERROR.
010166  005037  001232                 CLR     $TMP3
010172  010237  001234                 MOV     R2,$TMP4
010176  005037  001236                 CLR     $TMP5

010202  104121                 1$:     ERROR   121
010204  000402                 KF4:    BR      KF5

010206  000000                 KFTMP1: .WORD   0               ;USED TO DETERMINE THE TEST
010210  000000                 KFTMP2: .WORD   0               ;ADDRESSES.

010212                           KF5:                           ;DONE!

                                ;;*****************************************************************
                                ;*TEST 14        CACHE MAINTENANCE REGISTER COUNT PATTERN TEST
                                ;*
                                ;*THIS TEST RUNS A COUNT PATTERN THROUGH THE MAINTENANCE REGISTER'S
                                ;*BITS 15 TO 4. THIS IS DONE TO INSURE THAT THESE BITS ARE SETABLE
                                ;*AND THAT THE DATA PATH TO THE REGISTERS IS VIABLE. MISSES ARE FORCED
                                ;*TO BOTH GROUPS SO THAT NO CACHE DATA OR ADDRESS MEMORY
                                ;*ERRORS SHOULD OCCUR. ALSO ANY CYCLES DONE TO MAIN MEMORY
                                ;*ARE INSURED, BY PROPER SELECTION OF INSTRUCTIONS, TO RETURN
                                ;*DATA WITH THE PARITY BITS ON SO AS TO NOT CAUSE MAIN MEMORY PARITY
                                ;*ERRORS BY SETTING THE MAIN MEMORY MAINTENANCE FUNCTION WHICH WOULD
                                ;*EFFECTIVELY FORCE THE PARITY BITS READ FROM MAIN MEMORY TO A
                                ;*ONE. SINCE THESE PARITY ARE ALREADY ONES, NO ERRORS SHOULD OCCUR.
                                ;*
                                ;;*****************************************************************
010212  000004                 TST14:  SCOPE
010214  012737  000020  001274         MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000014          MA=$TN-1
                                                        ;SET THE SKAD REGISTER
010222  012737  010474  032100         MOV     #TST15,SKAD     ;IN CASE THE TEST ABORTS.

010230  113737  001102  001224         MOVB    $TSTNM,$TMP0

010236  104432                         SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
010240  104434                         SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
010242  012737  010376  000114         MOV     #MAERR,@#CACHVEC        ;IN CASE AN ERROR OCCURS WHILE
                                                        ;RUNNING A COUNT PATTERN
                                                        ;THROUGH THE MAINTENANCE
                                                        ;REGISTER SET UP THE PARITY ERROR
```

```
                                                                ;TRAP VECTOR; NOTE THAT NO ERRORS
                                                                ;SHOULD OCCUR IF THIS REGISTER
                                                                ;AND THE PARITY LOGIC IS FUNCTIONING
                                                                ;PROPERLY!
010250  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.

010256  012701  177750                  MOV     #MAINT,R1
010262  005004                          CLR     R4
010264  012737  010276  001110          MOV     #MA1,$LPERR
010272  012700  170000                  MOV     #170000,R0

010276  000240                  MA1:    NOP                     ;NOTE, THE CODE IN THIS ARE
010300  010411                          MOV     R4,(R1)         ;MA1 THROUGH MA2, ASSEMBLES TO
010302  011102                          MOV     (R1),R2         ;MACHINE CODE WHICH WILL
010304  005011                          CLR     (R1)            ;HAVE THE PARITY BITS ON, 1'S!
                                                                ;THE PATTERN IS LOADDED INTO THE
                                                                ;MAINETENANCE REGISTER, READ BACK
                                                                ;AND THE MAINTENANCE REGISTER
                                                                ;IS CLEARED.
010306  030011                          BIT     R0,(R1)         ;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.
010310  001402                          BEQ     .+6             ;IF SO, THEN ALL THAT CAN
010312  000000                          HALT                    ;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.
010314  000240                  MA2:    NOP

010316  011105                          MOV     (R1),R5         ;SEE IF ANY OF THE LOW ORDER
010320  001410                          BEQ     MA3             ;BITS, 11 THROUGH 0, ARE STUCK
                                                                ;AT ONE.
                                                                ;IF SO REPORT THE ERROR.
010322  010437  001230                  MOV     R4,$TMP2
010326  010537  001232                  MOV     R5,$TMP3
010332  104122                  1$:     ERROR   122
010334  012737  177777  032320          MOV     #-1,MANFLG      ;??????????????GO ON???????????

010342  020402                  MA3:    CMP     R4,R2           ;SEE IF THE PATTERN WRITTEN MATCHES
010344  001410                          BEQ     MA4             ;THE PATTERN READ.

                                                                ;IF NOT REPORT THE ERROR.
010346  010437  001230                  MOV     R4,$TMP2
010352  010237  001232                  MOV     R2,$TMP3
010356  104123                  1$:     ERROR   123
010360  012737  177777  032334          MOV     #-1,MANFL2

010366  062704  000020          MA4:    ADD     #20,R4          ;INCREMENT THE COUNT PATTERN.
010372  001341                          BNE     MA1
010374  000432                          BR      MADONE
```

```
2133
2134  010376                          MAERR:                          ;TRAP TO HERE IN THE EVENT
2135                                                                  ;THAT A PARITY ERROR OCCURS
2136                                                                  ;WHILE RUNNING THIS COUNT
2137                                                                  ;PATTERN TEST.
2138  010376  032737  000400  177744        BIT     #400,@#MEMERR   ;SEE IF THE ERROR WAS A MAINTENANCE
2139  010404  001005                        BNE     MAERR1          ;ERROR, CAUSED BY A MAINTENANCE
2140                                                                  ;FUNCTION. IF NOT GO TO THE
2141  010406  012737  031754  000114        MOV     #SPUR,@#CACHVEC ;SPUR ROUTINE WHICH HANDLES SUCH UNEXPECTED
2142
2143  010414  000137  031754              JMP     SPUR            ;ERRORS.
2144
2145  010420  013737  177744  001234  MAERR1: MOV     @#MEMERR,$TMP4  ;IF THE ERROR WAS CAUSED BY A
2146  010426  013737  177740  001226        MOV     @#LOADRS,$TMP1  ;MAINT FUNCTION THEN REPORT THE
2147  010434  013737  177742  001230        MOV     @#HIADRS,$TMP2  ;FAILURE OF THAT REGISTER.
2148  010442  012637  001232              MOV     (SP)+,$TMP3
2149  010446  005726                       TST     (SP)+
2150
2151  010450  104124                  1$:  ERROR   124
2152  010452  012737  177777  032334        MOV     #-1,MANFL2
2153
2154  010460  000742                        BR      MA4             ;RETURN TO THE TEST.
2155
2156  010462  005037  177746          MADONE: CLR     @#CONTRL        ;DONE
2157  010466  012737  031754  000114        MOV     #SPUR,@#CACHVEC
2158
2159
2160
2161
2162                                  ;;**************************************************************************
2163                                  ;*TEST 15       CACHE MAINTENANCE AND ERROR REGISTERS TEST 1
2164                                  ;*
2165                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY
2166                                  ;*ERROR ON THE MAIN MEMORY ADDRESS AND CONTROL LINES, AND ALSO A TEST
2167                                  ;*OF THE ERROR REGISTER'S ABILITY TO APPROPRIATELY SET TO 104402. THE
2168                                  ;*REFERENCE CAUSING THIS ERROR WILL BE MADE FROM THE CPU DIRECTLY TO
2169                                  ;*THE CACHE.
2170                                  ;*
2171                                  ;;**************************************************************************
2172  010474  000004                  TST15:  SCOPE
2173  010476  012737  000040  001274        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2174          000015                  MAB=$TN-1
2175                                                                  ;SET THE SKAD REGISTER
2176  010504  012737  010772  032100        MOV     #TST16,SKAD     ;IN CASE THE TEST ABORTS.
2177
2178  010512  113737  001102  001224        MOVB    $TSTNM,$TMP0
2179
2180  010520  104430                        SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2181  010522  104432                        SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2182  010524  104434                        SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2183  010526  104436                        SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2184  010530  012737  010600  000114        MOV     #MABRR0,@#CACHVEC       ;SET UP FOR THE ERROR.
2185
2186  010536  012704  000002                MOV     #2,R4           ;THIS IS THE PATTERN THAT WILL
2187  010542  012702  177750                MOV     #MAINT,R2       ;BE PUT IN THE MAINTENANCE REG.
2188  010546  012737  000014  177746        MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
```

```
2189
2190  010554  000240                        NOP                     ;FOR SCOPING.
2191  010556  010412                        MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
2192  010560  005012                        CLR     (R2)            ;THE REFERENCE WHICH FETCHES
2193                                                                 ;THIS INSTRUCTION SHOULD
2194                                                                 ;CAUSE THE ABORT!
2195
2196  010562                        MAB2:                           ;NO ABORT OCCURRED REPORT THE ERROR
2197  010562  010437  001230                MOV     R4,$TMP2
2198  010566  104127                1$:     ERROR   127
2199  010570  012737  177777  032334        MOV     #-1,MANFL2
2200  010576  000474                        BR      MABDON
2201
2202  010600  022737  104402  177744  MABRR0: CMP   #104402,@#MEMERR       ;WHEN THE TRAP IS MADE TO THIS LOCATION
2203  010606  001036                        BNE     MABRR4          ;MAKE SURE THE ERROR REGISTER IS
2204                                                                 ;SET CORRECTLY. IF NOT GO TO MABRR4.
2205  010610  022626                MABRR1: CMP     (SP)+,(SP)+     ;OTHERWISE RESET THE STACK.
2206  010612  012737  177777  177744  MABR15: MOV   #-1,@#MEMERR    ;ATTEMPT TO CLEAR THE ERROR REGISTER.
2207  010620  005737  177744                TST     @#MEMERR
2208  010624  001416                        BEQ     MABRR3
2209
2210  010626                        MABRR2:                         ;REPORT ERROR REGISTER WON'T CLEAR!
2211  010626  013737  177740  001230        MOV     @#LOADRS,$TMP2
2212  010634  013737  177742  001232        MOV     @#HIADRS,$TMP3
2213  010642  013737  177744  001234        MOV     @#MEMERR,$TMP4
2214  010650  104130                1$:     ERROR   130
2215  010652  012737  177777  032314        MOV     #-1,MMRFLG
2216  010660  000443                        BR      MABDON
2217
2218  010662  022737  177740  177740  MABRR3: CMP   #177740,@#LOADRS       ;MAKE SURE THE ADDRESS
2219  010670  001356                        BNE     MABRR2          ;REGISTER RESET.
2220  010672  022737  000003  177742        CMP     #3,@#HIADRS
2221  010700  001352                        BNE     MABRR2
2222  010702  000432                        BR      MABDON
2223
2224  010704                        MABRR4:                         ;REPORT ERROR REGISTER NOT SET CORRECTLY!!
2225  010704  012637  001230                MOV     (SP)+,$TMP2
2226  010710  005726                        TST     (SP)+
2227  010712  013737  177740  001232        MOV     @#LOADRS,$TMP3
2228  010720  013737  177742  001234        MOV     @#HIADRS,$TMP4
2229  010726  012737  000002  001236        MOV     #2,$TMP5
2230  010734  012737  104402  001240        MOV     #104402,$TMP6
2231  010742  013737  177744  001242        MOV     @#MEMERR,$TMP7
2232  010750  104131                1$:     ERROR   131
2233  010752  012737  177777  032334        MOV     #-1,MANFL2
2234  010760  012737  177777  032330        MOV     #-1,MMRFL2
2235  010766  000711                        BR      MABR15          ;GO SEE IF THE ERROR REGISTER
2236                                                                 ;CAN BE CLEARED.
2237  010770  104416                MABDON: RSET                    ;DONE!!
2238
2239
2240                                ;;*****************************************************************
2241                                ;*TEST 16       CACHE MAINTENANCE AND ERROR REGISTERS TEST 2
2242                                ;*
2243                                ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2244                                ;*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.
                                ;*
                                ;;*****************************************************************
010772  000004                TST16:  SCOPE
010774  012737  000040  001274        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000016                MB=$TN-1
                                                        ;SET THE SKAD REGISTER
011002  012737  011310  032100        MOV     #TST17,SKAD     ;IN CASE THE TEST ABORTS.

011010  113737  001102  001224        MOVB    $TSTNM,$TMP0

011016  104430                        SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
011020  104432                        SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
011022  104434                        SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
011024  104436                        SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
011026  012737  011106  000114        MOV     #MBERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
011034  012704  010000                MOV     #10000,R4       ;PATERN TO BE PUT INTO THE
011040  012702  177750                MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
011044  012737  000014  177746        MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
011052  000402                        BR      MB1

        011054                        LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        011054                        LOC=-4&LOC
        011060                        LOC=LOC+4
        011060                        .=LOC

011060  000240                MB1:    NOP
011062  010412                        MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
011064  005701                MB2:    TST     R1              ;THIS IS A DUMMY INSTRUCTION
                                                        ;WITH THE APPROPRIATE PARITY
                                                        ;WHOSE FETCH WILL CAUSE THE ERROR.
011066  005012                        CLR     (R2)

011070                          MB3:                            ;REPORT ERROR.  MAINTENANCE
011070  010437  001230                MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                        ;CAUSE ERROR.
011074  104127                  1$:   ERROR   127
011076  012737  177777  032334  MOV     #-1,MANFL2
011104  000500                        BR      MBDONE

011106  022737  104404  177744  MBERR0: CMP     #104404,@#MEMERR        ;DID THE ERROR REGISTER
011114  001042                        BNE     69$             ;SET PROPERLY?

011116  022626                  64$:  CMP     (SP)+,(SP)+     ;RESET THE STACK
011120  005037  177572          65$:  CLR     @#MMR0
011124  005037  172516                CLR     @#MMR3
011130  012737  177777  177744        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
011136  005737  177744                TST     @#MEMERR        ;REGISTER.
011142  001416                        BEQ     68$

011144                          66$:                            ;ERROR REGISTER WON'T
011144  013737  177740  001230        MOV     @#LOADRS,$TMP2  ;CLEAR
011152  013737  177742  001232        MOV     @#HIADRS,$TMP3
011160  013737  177744  001234        MOV     @#MEMERR,$TMP4

011166  104130                  67$:  ERROR   130
```

```
2301  011170  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2302  011176  000443                          BR      MBDONE
2303
2304  011200  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2305  011206  001356                          BNE     66$             ;UNLOCKED.
2306  011210  022737  000003  177742          CMP     #3,@#HIADRS
2307  011216  001352                          BNE     66$
2308  011220  000432                          BR      MBDONE
2309
2310  011222                          69$:                            ;REPORT ERROR REGISTER
2311  011222  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2312  011226  005726                          TST     (SP)+           ;RESET THE STACK.
2313  011230  013737  177740  001232          MOV     @#LOADRS,$TMP3
2314  011236  013737  177742  001234          MOV     @#HIADRS,$TMP4
2315  011244  012737  010000  001236          MOV     #10000,$TMP5
2316  011252  012737  104404  001240          MOV     #104404,$TMP6
2317  011260  013737  177744  001242          MOV     @#MEMERR,$TMP7
2318
2319  011266  104131                  70$:    ERROR   131
2320  011270  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2321  011276  012737  177777  032330          MOV     #-1,MMRFL2
2322  011304  000705                          BR      65$
2323  011306  104416                  MBDONE: RSET
2324
2325                                  ;;**********************************************************************
2326                                  ;*TEST 17        CACHE MAINTENANCE AND ERROR REGISTERS TEST 3
2327                                  ;*
2328                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2329                                  ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S HIGH BYTE,
2330                                  ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2331                                  ;*
2332                                  ;;**********************************************************************
2333  011310  000004                  TST17:  SCOPE
2334  011312  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2335           000017                 MC=$TN-1
2336                                                                  ;SET THE SKAD REGISTER
2337  011320  012737  011624  032100          MOV     #TST20,SKAD     ;IN CASE THE TEST ABORTS.
2338
2339  011326  113737  001102  001224          MOVB    $TSTNM,$TMP0
2340
2341  011334  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2342  011336  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2343  011340  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2344  011342  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2345  011344  012737  011422  000114          MOV     #MCERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
2346  011352  012704  020000                  MOV     #20000,R4       ;PATTERN TO BE USED IN THE
2347  011356  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
2348  011362  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
2349  011370  000401                          BR      MC1
2350
2351           011372                         LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2352           011370                         LOC=-4&LOC
2353           011374                         LOC=LOC+4
2354           011374                         .=LOC
2355
2356  011374  000240                  MC1:    NOP
```

```
2357  011376  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
2358  011400  005701                  MC2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
2359                                                                    ;SHOULD CAUSE THE ABORT.
2360  011402  005012                          CLR     (R2)
2361
2362  011404                           MC3:                            ;REPORT ERROR.  MAINTENANCE
2363  011404  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2364                                                                    ;CAUSE ERROR.
2365  011410  104127                  1$:     ERROR   127
2366  011412  012737  177777  032334  MOV     #-1,MANFL2
2367  011420  000500                          BR      MCDONE
2368
2369  011422  022737  104404  177744  MCERR0: CMP     #104404,@#MEMERR        ;DID THE ERROR REGISTER
2370  011430  001042                          BNE     69$             ;SET PROPERLY?
2371
2372  011432  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2373  011434  005037  177572          65$:    CLR     @#MMR0
2374  011440  005037  172516                  CLR     @#MMR3
2375  011444  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2376  011452  005737  177744                  TST     @#MEMERR        ;REGISTER.
2377  011456  001416                          BEQ     68$
2378
2379  011460                           66$:                            ;ERROR REGISTER WON'T
2380  011460  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
2381  011466  013737  177742  001232          MOV     @#HIADRS,$TMP3
2382  011474  013737  177744  001234          MOV     @#MEMERR,$TMP4
2383
2384  011502  104130                  67$:    ERROR   130
2385  011504  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2386  011512  000443                          BR      MCDONE
2387
2388  011514  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2389  011522  001356                          BNE     66$             ;UNLOCKED.
2390  011524  022737  000003  177742          CMP     #3,@#HIADRS
2391  011532  001352                          BNE     66$
2392  011534  000432                          BR      MCDONE
2393
2394  011536                           69$:                            ;REPORT ERROR REGISTER
2395  011536  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2396  011542  005726                          TST     (SP)+           ;RESET THE STACK.
2397  011544  013737  177740  001232          MOV     @#LOADRS,$TMP3
2398  011552  013737  177742  001234          MOV     @#HIADRS,$TMP4
2399  011560  012737  020000  001236          MOV     #20000,$TMP5
2400  011566  012737  104404  001240          MOV     #104404,$TMP6
2401  011574  013737  177744  001242          MOV     @#MEMERR,$TMP7
2402
2403  011602  104131                  70$:    ERROR   131
2404  011604  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2405  011612  012737  177777  032330          MOV     #-1,MMRFL2
2406  011620  000705                          BR      65$
2407  011622  104416                  MCDONE: RSET
2408
2409                                    ;;*****************************************************************************
2410                                    ;*TEST 20        CACHE MAINTENANCE AND ERROR REGISTERS TEST 4
2411                                    ;*
2412                                    ;*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.
                                        ;*
                                        ;;*****************************************************************************
011624  000004                        TST20:  SCOPE
011626  012737  000040  001274                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000020                  MD=$TN-1
                                                                        ;SET THE SKAD REGISTER
011634  012737  012144  032100                MOV     #TST21,SKAD     ;IN CASE THE TEST ABORTS.

011642  113737  001102  001224                MOVB    $TSTNM,$TMP0

011650  104430                                SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
011652  104432                                SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
011654  104434                                SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
011656  104436                                SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
011660  012737  011742  000114                MOV     #MDERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
011666  012704  040000                        MOV     #40000,R4       ;PATTERN TO BE PUT IN THE
011672  012702  177750                        MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
011676  012737  000014  177746                MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
011704  000402                                BR      MD1

        011706                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        011704                          LOC=-4&LOC
        011710                          LOC=LOC+4
        011710                          .=LOC

011710  000240                                NOP
011712  000240                        MD1:    NOP
011714  010412                                MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
011716  005701                        MD2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
                                                                        ;SHOULD CAUSE THE MAIN MEMORY
                                                                        ;DATA PARITY ABORT.
011720  005012                                CLR     (R2)
011722  000240                                NOP

011724                                  MD3:                            ;REPORT ERROR.  MAINTENANCE
011724  010437  001230                        MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                        ;CAUSE ERROR.
011730  104127                          1$:     ERROR   127
011732  012737  177777  032334        MOV     #-1,MANFL2
011740  000500                                BR      MDDONE

011742  022737  104410  177744        MDERR0: CMP     #104410,@#MEMERR        ;DID THE ERROR REGISTER
011750  001042                                BNE     69$             ;SET PROPERLY?

011752  022626                          64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
011754  005037  177572                  65$:    CLR     @#MMR0
011760  005037  172516                          CLR     @#MMR3
011764  012737  177777  177744                MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
011772  005737  177744                        TST     @#MEMERR        ;REGISTER.
011776  001416                                BEQ     68$

012000                                  66$:                            ;ERROR REGISTER WON'T
012000  013737  177740  001230                MOV     @#LOADRS,$TMP2  ;CLEAR
012006  013737  177742  001232                MOV     @#HIADRS,$TMP3
```

```
2469  012014  013737  177744  001234          MOV     @#MEMERR,$TMP4
2470
2471  012022  104130                  67$:    ERROR   130
2472  012024  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2473  012032  000443                          BR      MDDONE
2474
2475  012034  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2476  012042  001356                          BNE     66$             ;UNLOCKED.
2477  012044  022737  000003  177742          CMP     #3,@#HIADRS
2478  012052  001352                          BNE     66$
2479  012054  000432                          BR      MDDONE
2480
2481  012056                          69$:                            ;REPORT ERROR REGISTER
2482  012056  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2483  012062  005726                          TST     (SP)+           ;RESET THE STACK.
2484  012064  013737  177740  001232          MOV     @#LOADRS,$TMP3
2485  012072  013737  177742  001234          MOV     @#HIADRS,$TMP4
2486  012100  012737  040000  001236          MOV     #40000,$TMP5
2487  012106  012737  104410  001240          MOV     #104410,$TMP6
2488  012114  013737  177744  001242          MOV     @#MEMERR,$TMP7
2489
2490  012122  104131                  70$:    ERROR   131
2491  012124  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2492  012132  012737  177777  032330          MOV     #-1,MMRFL2
2493  012140  000705                          BR      65$
2494  012142  104416          MDDONE: RSET
2495
2496                                  ;;**********************************************************************
2497                                  ;*TEST 21        CACHE MAINTENANCE AND ERROR REGISTERS TEST 5
2498                                  ;*
2499                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2500                                  ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S HIGH BYTE,
2501                                  ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2502                                  ;*
2503                                  ;;**********************************************************************
2504  012144  000004          TST21:  SCOPE
2505  012146  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2506          000021          ME=$TN-1
2507                                                                  ;SET THE SKAD REGISTER
2508  012154  012737  012464  032100          MOV     #TST22,SKAD     ;IN CASE THE TEST ABORTS.
2509
2510  012162  113737  001102  001224          MOVB    $TSTNM,$TMP0
2511
2512  012170  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2513  012172  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2514  012174  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2515  012176  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2516  012200  012737  012262  000114          MOV     #MEERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
2517  012206  012704  100000                  MOV     #100000,R4      ;PATTERN TO BE PUT IN THE
2518  012212  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
2519  012216  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
2520  012224  000402                          BR      ME1
2521
2522          012226                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2523          012224                          LOC=-4&LOC
2524          012230                          LOC=LOC+4
```

```
2525            012230                          .=LOC
2526
2527  012230    000240                          NOP
2528  012232    000240                  ME1:    NOP
2529  012234    010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
2530  012236    005701                  ME2:    TST     R1              ;THE FETCH OF THIS INSTRUSTION
2531                                                                    ;SHOULD CAUSE THE ABORT.
2532  012240    005012                          CLR     (R2)
2533  012242    000240                          NOP
2534
2535  012244                            ME3:                            ;REPORT ERROR.  MAINTENANCE
2536  012244    010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2537                                                                    ;CAUSE ERROR.
2538  012250    104127                  1$:     ERROR   127
2539  012252    012737  177777  032334  MOV     #-1,MANFL2
2540  012260    000500                          BR      MEDONE
2541
2542  012262    022737  104410  177744  MEERR0: CMP     #104410,@#MEMERR        ;DID THE ERROR REGISTER
2543  012270    001042                          BNE     69$             ;SET PROPERLY?
2544
2545  012272    022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2546  012274    005037  177572          65$:    CLR     @#MMR0
2547  012300    005037  172516                  CLR     @#MMR3
2548  012304    012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2549  012312    005737  177744                  TST     @#MEMERR        ;REGISTER.
2550  012316    001416                          BEQ     68$
2551
2552  012320                            66$:                            ;ERROR REGISTER WON'T
2553  012320    013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
2554  012326    013737  177742  001232          MOV     @#HIADRS,$TMP3
2555  012334    013737  177744  001234          MOV     @#MEMERR,$TMP4
2556
2557  012342    104130                  67$:    ERROR   130
2558  012344    012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2559  012352    000443                          BR      MEDONE
2560
2561  012354    022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2562  012362    001356                          BNE     66$             ;UNLOCKED.
2563  012364    022737  000003  177742          CMP     #3,@#HIADRS
2564  012372    001352                          BNE     66$
2565  012374    000432                          BR      MEDONE
2566
2567  012376                            69$:                            ;REPORT ERROR REGISTER
2568  012376    012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2569  012402    005726                          TST     (SP)+           ;RESET THE STACK.
2570  012404    013737  177740  001232          MOV     @#LOADRS,$TMP3
2571  012412    013737  177742  001234          MOV     @#HIADRS,$TMP4
2572  012420    012737  100000  001236          MOV     #100000,$TMP5
2573  012426    012737  104410  001240          MOV     #104410,$TMP6
2574  012434    013737  177744  001242          MOV     @#MEMERR,$TMP7
2575
2576  012442    104131                  70$:    ERROR   131
2577  012444    012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2578  012452    012737  177777  032330          MOV     #-1,MMRFL2
2579  012460    000705                          BR      65$
2580  012462    104416                  MEDONE: RSET
```

```
                                ;;*********************************************************************
                                ;*TEST 22       CACHE MAINTENANCE AND ERROR REGISTERS TEST 6
                                ;*
                                ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
                                ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE,
                                ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
                                ;*
                                ;;*********************************************************************
012464  000004                TST22:  SCOPE
012466  012737  000040  001274        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000022                MF=$TN-1
                                                            ;SET THE SKAD REGISTER
012474  012737  013000  032100        MOV     #TST23,SKAD     ;IN CASE THE TEST ABORTS.

012502  113737  001102  001224        MOVB    $TSTNM,$TMP0
012510  012737  012576  000114        MOV     #MFERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
012516  012704  010000                MOV     #10000,R4       ;PATTERN TO BE LOADED INTO THE
012522  012702  177750                MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
012526  012737  000014  177746        MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
012534  012705  012556                MOV     #MF2,R5         ;A REFERENCE TO THIS ADDRESS
                                                            ;WILL CAUSE A PARITY TRAP BECAUSE
                                                            ;THE OTHER WORD IN THE PAIR
                                                            ;WILL HAVE THE APPROPRIATE
                                                            ;PARITY TO CAUSE THE MAINTENANCE
                                                            ;FUNCTION WHICH WILL BE SET
                                                            ;TO FORCE THE ERROR.
012540  000401                        BR      MF1

        012542                        LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        012540                        LOC=-4&LOC
        012544                        LOC=LOC+4
        012544                        .=LOC

012544  000240                MF1:    NOP
012546  010412                        MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
012550  021502                        CMP     (R5),R2         ;THIS REFERENCE TO (R5) WILL CAUSE A
012552  005012                        CLR     (R2)            ;PARITY TRAP SINCE THE OTHER IN THAT
                                                            ;PAIR WILL CAUSE A PARITY ERROR.
012554  005701                        TST     R1              ;THIS WORD WILL CAUSE THE ERROR
012556  000240                MF2:    NOP                     ;WHEN THIS WORD IS REFERENCED.

012560                        MF3:                            ;REPORT ERROR.  MAINTENANCE
012560  010437  001230                MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                            ;CAUSE ERROR.
012564  104127                1$:     ERROR   127
012566  012737  177777  032334  MOV   #-1,MANFL2
012574  000500                        BR      MFDONE

012576  022737  004404  177744  MFERR0: CMP     #4404,@#MEMERR  ;DID THE ERROR REGISTER
012604  001042                        BNE     69$             ;SET PROPERLY?

012606  022626                64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
012610  005037  177572        65$:    CLR     @#MMR0
012614  005037  172516                CLR     @#MMR3
012620  012737  177777  177744        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
```

```
2637  012626  005737  177744          TST     @#MEMERR        ;REGISTER.
2638  012632  001416                  BEQ     68$
2639
2640  012634                  66$:                            ;ERROR REGISTER WON'T
2641  012634  013737  177740  001230  MOV     @#LOADRS,$TMP2  ;CLEAR
2642  012642  013737  177742  001232  MOV     @#HIADRS,$TMP3
2643  012650  013737  177744  001234  MOV     @#MEMERR,$TMP4
2644
2645  012656  104130          67$:    ERROR   130
2646  012660  012737  177777  032314  MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2647  012666  000443                  BR      MFDONE
2648
2649  012670  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2650  012676  001356                  BNE     66$             ;UNLOCKED.
2651  012700  022737  000003  177742  CMP     #3,@#HIADRS
2652  012706  001352                  BNE     66$
2653  012710  000432                  BR      MFDONE
2654
2655  012712                  69$:                            ;REPORT ERROR REGISTER
2656  012712  012637  001230          MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2657  012716  005726                  TST     (SP)+           ;RESET THE STACK.
2658  012720  013737  177740  001232  MOV     @#LOADRS,$TMP3
2659  012726  013737  177742  001234  MOV     @#HIADRS,$TMP4
2660  012734  012737  010000  001236  MOV     #10000,$TMP5
2661  012742  012737  004404  001240  MOV     #4404,$TMP6
2662  012750  013737  177744  001242  MOV     @#MEMERR,$TMP7
2663
2664  012756  104131          70$:    ERROR   131
2665  012760  012737  177777  032334  MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2666  012766  012737  177777  032330  MOV     #-1,MMRFL2
2667  012774  000705                  BR      65$
2668  012776  104416          MFDONE: RSET
2669
2670                                  ;;*****************************************************************
2671                                  ;*TEST 23       CACHE MAINTENANCE AND ERROR REGISTERS TEST 7
2672                                  ;*
2673                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2674                                  ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE,
2675                                  ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2676                                  ;*
2677                                  ;;*****************************************************************
2678  013000  000004          TST23:  SCOPE
2679  013002  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2680          000023          MG=$TN-1
2681                                                          ;SET THE SKAD REGISTER
2682  013010  012737  013320  032100          MOV     #TST24,SKAD     ;IN CASE THE TEST ABORTS.
2683
2684  013016  113737  001102  001224          MOVB    $TSTNM,$TMP0
2685
2686  013024  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2687  013026  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2688  013030  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2689  013032  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2690  013034  012704  040000                  MOV     #40000,R4       ;THIS PATTERN WILL BE PUT IN THE
2691  013040  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
2692  013044  012737  013116  000114          MOV     #MGERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
```

```
2693  013052  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
2694  013060  000401                          BR      MG1
2695
2696          013062                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2697          013060                          LOC=-4&LOC
2698          013064                          LOC=LOC+4
2699          013064                          .=LOC
2700
2701  013064  000240                  MG1:    NOP
2702  013066  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
2703  013070  000240                          NOP                     ;THE REFERENCE TO THIS NOP
2704  013072  005701                  MG2:    TST     R1              ;SHOULD CAUSE A PARITY ERROR TO OCCUR AT
2705                                                                  ;MG2, RESULTING IN A TRAP!
2706  013074  005012                          CLR     (R2)
2707  013076  000240                          NOP
2708
2709  013100                          MG3:                            ;REPORT ERROR.  MAINTENANCE
2710  013100  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2711                                                                  ;CAUSE ERROR.
2712  013104  104127                  1$:     ERROR   127
2713  013106  012737  177777  032334  MOV     #-1,MANFL2
2714  013114  000500                          BR      MGDONE
2715
2716  013116  022737  004410  177744  MGERR0: CMP     #4410,@#MEMERR  ;DID THE ERROR REGISTER
2717  013124  001042                          BNE     69$             ;SET PROPERLY?
2718
2719  013126  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2720  013130  005037  177572          65$:    CLR     @#MMR0
2721  013134  005037  172516                  CLR     @#MMR3
2722  013140  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2723  013146  005737  177744                  TST     @#MEMERR        ;REGISTER.
2724  013152  001416                          BEQ     68$
2725
2726  013154                          66$:                            ;ERROR REGISTER WON'T
2727  013154  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
2728  013162  013737  177742  001232          MOV     @#HIADRS,$TMP3
2729  013170  013737  177744  001234          MOV     @#MEMERR,$TMP4
2730
2731  013176  104130                  67$:    ERROR   130
2732  013200  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2733  013206  000443                          BR      MGDONE
2734
2735  013210  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2736  013216  001356                          BNE     66$             ;UNLOCKED.
2737  013220  022737  000003  177742          CMP     #3,@#HIADRS
2738  013226  001352                          BNE     66$
2739  013230  000432                          BR      MGDONE
2740
2741  013232                          69$:                            ;REPORT ERROR REGISTER
2742  013232  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2743  013236  005726                          TST     (SP)+           ;RESET THE STACK.
2744  013240  013737  177740  001232          MOV     @#LOADRS,$TMP3
2745  013246  013737  177742  001234          MOV     @#HIADRS,$TMP4
2746  013254  012737  040000  001236          MOV     #40000,$TMP5
2747  013262  012737  004410  001240          MOV     #4410,$TMP6
2748  013270  013737  177744  001242          MOV     @#MEMERR,$TMP7
```

```
                                        .
013276  104131                  70$:    ERROR   131
013300  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
013306  012737  177777  032330          MOV     #-1,MMRFL2
013314  000705                          BR      65$
013316  104416                  MGDONE: RSET


                                ;;*******************************************************************
                                ;*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.
                                ;*
                                ;;*******************************************************************
013320  000004                  TST24:  SCOPE
013322  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000024                  MH=$TN-1
                                                                ;SET THE SKAD REGISTER
013330  012737  013664  032100          MOV     #TST25,SKAD     ;IN CASE THE TEST ABORTS.

013336  113737  001102  001224          MOVB    $TSTNM,$TMP0

013344  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
013346  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
013350  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
013352  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
013354  012737  013462  000114          MOV     #MHERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
013362  012704  000400                  MOV     #400,R4 ;PATTERN TO BE PUT IN MAINT. REG.
013366  012702  177750                  MOV     #MAINT,R2
013372  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
                                                                ;FORCE MISS THE OTHER
                                                                ;GROUP
013400  012705  013442                  MOV     #MH1,R5         ;MAKE MH1 A HIT IN
013404  005715                          TST     (R5)            ;GROUP GP.
013406  005715                          TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
013410  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
013416  001007                          BNE     1$
                                                                ;IF NOT ERROR!
013420  010537  001230                  MOV     R5,$TMP2
013424  012737  000000  001226          MOV     #0,$TMP1
013432  104001                          ERROR   1

013434  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

013436  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
013440  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
013442  005012                  MH1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
```

```
                                                        ;CACHE ADDRESS MEMORY GROUP GP.


013444                          MH2:                    ;REPORT ERROR.  MAINTENANCE
013444  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                        ;CAUSE ERROR.
013450  104127                  1$:     ERROR   127
013452  012737  177777  032334  MOV     #-1,MANFL2
013460  000500                  BR      MHDONE

013462  022737  004420  177744  MHERR0: CMP     #4420,@#MEMERR  ;DID THE ERROR REGISTER
013470  001042                  BNE     69$             ;SET PROPERLY?

013472  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
013474  005037  177572          65$:    CLR     @#MMR0
013500  005037  172516                  CLR     @#MMR3
013504  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
013512  005737  177744                  TST     @#MEMERR        ;REGISTER.
013516  001416                  BEQ     68$

013520                          66$:                    ;ERROR REGISTER WON'T
013520  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
013526  013737  177742  001232          MOV     @#HIADRS,$TMP3
013534  013737  177744  001234          MOV     @#MEMERR,$TMP4

013542  104130                  67$:    ERROR   130
013544  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
013552  000443                  BR      MHDONE

013554  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
013562  001356                  BNE     66$             ;UNLOCKED.
013564  022737  000003  177742          CMP     #3,@#HIADRS
013572  001352                  BNE     66$
013574  000432                  BR      MHDONE

013576                          69$:                    ;REPORT ERROR REGISTER
013576  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
013602  005726                  TST     (SP)+           ;RESET THE STACK.
013604  013737  177740  001232          MOV     @#LOADRS,$TMP3
013612  013737  177742  001234          MOV     @#HIADRS,$TMP4
013620  012737  000400  001236          MOV     #400,$TMP5
013626  012737  004420  001240          MOV     #4420,$TMP6
013634  013737  177744  001242          MOV     @#MEMERR,$TMP7

013642  104131                  70$:    ERROR   131
013644  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
013652  012737  177777  032330          MOV     #-1,MMRFL2
013660  000705                  BR      65$
013662  104416                  MHDONE: RSET


;;*******************************************************************
;*TEST 25        CACHE MAINTENANCE AND ERROR REGISTERS TEST 11
;*
;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO. FOR THE
```

```
                                ;*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
                                ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
                                ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
                                ;*TO THE CACHE.
                                ;*
                                ;;*****************************************************************
013664  000004                  TST25:  SCOPE
013666  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000025                  MI=$TN-1
                                                                ;SET THE SKAD REGISTER
013674  012737  014230  032100          MOV     #TST26,SKAD     ;IN CASE THE TEST ABORTS.

013702  113737  001102  001224          MOVB    $TSTNM,$TMP0

013710  104430                         SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
013712  104432                         SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
013714  104434                         SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
013716  104436                         SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
013720  012737  014026  000114          MOV     #MIERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
013726  012704  001000                  MOV     #1000,R4        ;PATTERN TO BE PUT IN MAINT. REG.
013732  012702  177750                  MOV     #MAINT,R2
013736  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
                                                                ;FORCE MISS THE OTHER
                                                                ;GROUP
013744  012705  014006                  MOV     #MI1,R5         ;MAKE MI1 A HIT IN
013750  005715                          TST     (R5)            ;GROUP GP.
013752  005715                          TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
013754  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
013762  001007                          BNE     1$
                                                                ;IF NOT ERROR!
013764  010537  001230                  MOV     R5,$TMP2
013770  012737  000000  001226          MOV     #0,$TMP1
013776  104001                          ERROR   1

014000  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

014002  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
014004  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
014006  005012                  MI1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
                                                                ;CACHE ADDRESS MEMORY GROUP GP.


014010                          MI2:                            ;REPORT ERROR.  MAINTENANCE
014010  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
014014  104127                  1$:     ERROR   127
014016  012737  177777  032334  MOV     #-1,MANFL2
014024  000500                          BR      MIDONE

014026  022737  004420  177744  MIERR0: CMP     #4420,@#MEMERR  ;DID THE ERROR REGISTER
014034  001042                          BNE     69$             ;SET PROPERLY?
```

```
2917  014036  022626                    64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2918  014040  005037  177572            65$:    CLR     @#MMR0
2919  014044  005037  172516                    CLR     @#MMR3
2920  014050  012737  177777  177744           MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2921  014056  005737  177744                    TST     @#MEMERR        ;REGISTER.
2922  014062  001416                           BEQ     68$
2923
2924  014064                            66$:                            ;ERROR REGISTER WON'T
2925  014064  013737  177740  001230           MOV     @#LOADRS,$TMP2  ;CLEAR
2926  014072  013737  177742  001232           MOV     @#HIADRS,$TMP3
2927  014100  013737  177744  001234           MOV     @#MEMERR,$TMP4
2928
2929  014106  104130                    67$:    ERROR   130
2930  014110  012737  177777  032314           MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2931  014116  000443                           BR      MIDONE
2932
2933  014120  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2934  014126  001356                           BNE     66$             ;UNLOCKED.
2935  014130  022737  000003  177742           CMP     #3,@#HIADRS
2936  014136  001352                           BNE     66$
2937  014140  000432                           BR      MIDONE
2938
2939  014142                            69$:                            ;REPORT ERROR REGISTER
2940  014142  012637  001230                    MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2941  014146  005726                           TST     (SP)+           ;RESET THE STACK.
2942  014150  013737  177740  001232           MOV     @#LOADRS,$TMP3
2943  014156  013737  177742  001234           MOV     @#HIADRS,$TMP4
2944  014164  012737  001000  001236           MOV     #1000,$TMP5
2945  014172  012737  004420  001240           MOV     #4420,$TMP6
2946  014200  013737  177744  001242           MOV     @#MEMERR,$TMP7
2947
2948  014206  104131                    70$:    ERROR   131
2949  014210  012737  177777  032334           MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2950  014216  012737  177777  032330           MOV     #-1,MMRFL2
2951  014224  000705                           BR      65$
2952  014226  104416                    MIDONE: RSET
2953
2954
2955                                    ;;*******************************************************************
2956                                    ;*TEST 26        CACHE MAINTENANCE AND ERROR REGISTERS TEST 12
2957                                    ;*
2958                                    ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
2959                                    ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE
2960                                    ;*LOW BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
2961                                    ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
2962                                    ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
2963                                    ;*TO THE CACHE.
2964                                    ;*
2965                                    ;;*******************************************************************
2966  014230  000004                    TST26:  SCOPE
2967  014232  012737  000040  001274           MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2968          000026                    MJ=$TN-1
2969                                                                    ;SET THE SKAD REGISTER
2970  014240  012737  014574  032100           MOV     #TST27,SKAD     ;IN CASE THE TEST ABORTS.
2971
2972  014246  113737  001102  001224           MOVB    $TSTNM,$TMP0
```

```
2973
2974  014254  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2975  014256  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2976  014260  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2977  014262  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2978  014264  012737  014372  000114          MOV     #MJERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
2979  014272  012704  002000                  MOV     #2000,R4        ;PATTERN TO BE PUT IN MAINT. REG.
2980  014276  012702  177750                  MOV     #MAINT,R2
2981  014302  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
2982                                                                  ;FORCE MISS THE OTHER
2983                                                                  ;GROUP
2984  014310  012705  014352                  MOV     #MJ1,R5         ;MAKE MJ1 A HIT IN
2985  014314  005715                          TST     (R5)            ;GROUP GP.
2986  014316  005715                          TST     (R5)
2987
2988                                                                  ;SEE IF REFERENCE ADDRESS
2989  014320  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
2990  014326  001007                          BNE     1$
2991                                                                  ;IF NOT ERROR!
2992  014330  010537  001230                  MOV     R5,$TMP2
2993  014334  012737  000001  001226          MOV     #1,$TMP1
2994  014342  104001                          ERROR   1
2995
2996  014344  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
2997
2998  014346  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
2999  014350  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
3000  014352  005012                  MJ1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
3001                                                                  ;INSTRUCTION SHOULD CAUSE
3002                                                                  ;A PARITY ERROR IN THE
3003                                                                  ;CACHE. ADDRESS MEMORY GROUP GP.
3004
3005
3006  014354                          MJ2:                            ;REPORT ERROR.  MAINTENANCE
3007  014354  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
3008                                                                  ;CAUSE ERROR.
3009  014360  104127                  1$:     ERROR   127
3010  014362  012737  177777  032334  MOV     #-1,MANFL2
3011  014370  000500                          BR      MJDONE
3012
3013  014372  022737  004440  177744  MJERR0: CMP     #4440,@#MEMERR  ;DID THE ERROR REGISTER
3014  014400  001042                          BNE     69$             ;SET PROPERLY?
3015
3016  014402  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3017  014404  005037  177572          65$:    CLR     @#MMR0
3018  014410  005037  172516                  CLR     @#MMR3
3019  014414  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3020  014422  005737  177744                  TST     @#MEMERR        ;REGISTER.
3021  014426  001416                          BEQ     68$
3022
3023  014430                          66$:                            ;ERROR REGISTER WON'T
3024  014430  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
3025  014436  013737  177742  001232          MOV     @#HIADRS,$TMP3
3026  014444  013737  177744  001234          MOV     @#MEMERR,$TMP4
3027
3028  014452  104130                  67$:    ERROR   130
```

```
3029  014454  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3030  014462  000443                          BR      MJDONE
3031
3032  014464  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3033  014472  001356                          BNE     66$             ;UNLOCKED.
3034  014474  022737  000003  177742          CMP     #3,@#HIADRS
3035  014502  001352                          BNE     66$
3036  014504  000432                          BR      MJDONE
3037
3038  014506                          69$:                            ;REPORT ERROR REGISTER
3039  014506  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3040  014512  005726                          TST     (SP)+           ;RESET THE STACK.
3041  014514  013737  177740  001232          MOV     @#LOADRS,$TMP3
3042  014522  013737  177742  001234          MOV     @#HIADRS,$TMP4
3043  014530  012737  002000  001236          MOV     #2000,$TMP5
3044  014536  012737  004440  001240          MOV     #4440,$TMP6
3045  014544  013737  177744  001242          MOV     @#MEMERR,$TMP7
3046
3047  014552  104131                  70$:    ERROR   131
3048  014554  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3049  014562  012737  177777  032330          MOV     #-1,MMRFL2
3050  014570  000705                          BR      65$
3051  014572  104416                  MJDONE: RSET
3052
3053
3054                                  ;;**********************************************************************
3055                                  ;*TEST 27        CACHE MAINTENANCE AND ERROR REGISTERS TEST 13
3056                                  ;*
3057                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3058                                  ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE
3059                                  ;*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
3060                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3061                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3062                                  ;*TO THE CACHE.
3063                                  ;*
3064                                  ;;**********************************************************************
3065  014574  000004                  TST27:  SCOPE
3066  014576  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3067          000027          MK=$TN-1
3068                                                                  ;SET THE SKAD REGISTER
3069  014604  012737  015140  032100          MOV     #TST30,SKAD     ;IN CASE THE TEST ABORTS.
3070
3071  014612  113737  001102  001224          MOVB    $TSTNM,$TMP0
3072
3073  014620  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3074  014622  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3075  014624  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3076  014626  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3077  014630  012737  014736  000114          MOV     #MKERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
3078  014636  012704  004000                  MOV     #4000,R4        ;PATTERN TO BE PUT IN MAINT. REG.
3079  014642  012702  177750                  MOV     #MAINT,R2
3080  014646  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
3081                                                                  ;FORCE MISS THE OTHER
3082                                                                  ;GROUP
3083  014654  012705  014716                  MOV     #MK1,R5         ;MAKE MK1 A HIT IN
3084  014660  005715                          TST     (R5)            ;GROUP GP.
```

```
3085  014662  005715                          TST     (R5)
3086
3087                                                                ;SEE IF REFERENCE ADDRESS
3088  014664  032737  000010  177752          BIT     #10,@#HITMIS  ;IS A HIT.
3089  014672  001007                          BNE     1$
3090                                                                ;IF NOT ERROR!
3091  014674  010537  001230                  MOV     R5,$TMP2
3092  014700  012737  000001  001226          MOV     #1,$TMP1
3093  014706  104001                          ERROR   1
3094
3095  014710  104420                          SKIPT                 ;ERROR FATAL. GO TO NEXT TEST.
3096
3097  014712  000240                  1$:     NOP                   ;PUT THE PATTERN IN THE
3098  014714  010412                          MOV     R4,(R2)       ;MAINTENANCE REGISTER.
3099  014716  005012                  MK1:    CLR     (R2)          ;THE FETCH OF THIS NEXT
3100                                                                ;INSTRUCTION SHOULD CAUSE
3101                                                                ;A PARITY ERROR IN THE
3102                                                                ;CACHE ADDRESS MEMORY GROUP GP.
3103
3104
3105  014720                          MK2:                          ;REPORT ERROR.  MAINTENANCE
3106  014720  010437  001230                  MOV     R4,$TMP2      ;FUNCTION FAILED TO
3107                                                                ;CAUSE ERROR.
3108  014724  104127                  1$:     ERROR   127
3109  014726  012737  177777  032334  MOV     #-1,MANFL2
3110  014734  000500                          BR      MKDONE
3111
3112  014736  022737  004440  177744  MKERR0: CMP     #4440,@#MEMERR ;DID THE ERROR REGISTER
3113  014744  001042                          BNE     69$           ;SET PROPERLY?
3114
3115  014746  022626                  64$:    CMP     (SP)+,(SP)+   ;RESET THE STACK
3116  014750  005037  177572          65$:    CLR     @#MMR0
3117  014754  005037  172516                  CLR     @#MMR3
3118  014760  012737  177777  177744          MOV     #-1,@#MEMERR  ;TRY TO CLEAR THE ERROR
3119  014766  005737  177744                  TST     @#MEMERR      ;REGISTER.
3120  014772  001416                          BEQ     68$
3121
3122  014774                          66$:                          ;ERROR REGISTER WON'T
3123  014774  013737  177740  001230          MOV     @#LOADRS,$TMP2 ;CLEAR
3124  015002  013737  177742  001232          MOV     @#HIADRS,$TMP3
3125  015010  013737  177744  001234          MOV     @#MEMERR,$TMP4
3126
3127  015016  104130                  67$:    ERROR   130
3128  015020  012737  177777  032314          MOV     #-1,MMRFLG    ;SIGNAL BAD REGISTER
3129  015026  000443                          BR      MKDONE
3130
3131  015030  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3132  015036  001356                          BNE     66$           ;UNLOCKED.
3133  015040  022737  000003  177742          CMP     #3,@#HIADRS
3134  015046  001352                          BNE     66$
3135  015050  000432                          BR      MKDONE
3136
3137  015052                          69$:                          ;REPORT ERROR REGISTER
3138  015052  012637  001230                  MOV     (SP)+,$TMP2   ;NOT SET AS EXPECTED.
3139  015056  005726                          TST     (SP)+         ;RESET THE STACK.
3140  015060  013737  177740  001232          MOV     @#LOADRS,$TMP3
```

```
3141  015066  013737  177742  001234          MOV     @#HIADRS,$TMP4
3142  015074  012737  004000  001236          MOV     #4000,$TMP5
3143  015102  012737  004440  001240          MOV     #4440,$TMP6
3144  015110  013737  177744  001242          MOV     @#MEMERR,$TMP7
3145
3146  015116  104131                  70$:    ERROR   131
3147  015120  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3148  015126  012737  177777  032330          MOV     #-1,MMRFL2
3149  015134  000705                          BR      65$
3150  015136  104416                  MKDONE: RSET
3151
3152
3153                                  ;;**********************************************************************
3154                                  ;*TEST 30       CACHE MAINTENANCE AND ERROR REGISTERS TEST 14
3155                                  ;*
3156                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3157                                  ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ZERO, FOR THE
3158                                  ;*LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3159                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3160                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3161                                  ;*TO THE CACHE.
3162                                  ;*
3163                                  ;;**********************************************************************
3164  015140  000004          TST30:  SCOPE
3165  015142  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3166          000030          ML=$TN-1
3167                                                                  ;SET THE SKAD REGISTER
3168  015150  012737  015504  032100          MOV     #TST31,SKAD     ;IN CASE THE TEST ABORTS.
3169
3170  015156  113737  001102  001224          MOVB    $TSTNM,$TMP0
3171
3172  015164  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3173  015166  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3174  015170  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3175  015172  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3176  015174  012737  015302  000114          MOV     #MLERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
3177  015202  012704  000020                  MOV     #20,R4  ;PATTERN TO BE PUT IN MAINT. REG.
3178  015206  012702  177750                  MOV     #MAINT,R2
3179  015212  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
3180                                                                  ;FORCE MISS THE OTHER
3181                                                                  ;GROUP
3182  015220  012705  015262                  MOV     #ML1,R5         ;MAKE ML1 A HIT IN
3183  015224  005715                          TST     (R5)            ;GROUP GP.
3184  015226  005715                          TST     (R5)
3185
3186                                                                  ;SEE IF REFERENCE ADDRESS
3187  015230  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
3188  015236  001007                          BNE     1$
3189                                                                  ;IF NOT ERROR!
3190  015240  010537  001230                  MOV     R5,$TMP2
3191  015244  012737  000000  001226          MOV     #0,$TMP1
3192  015252  104001                          ERROR   1
3193
3194  015254  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
3195
3196  015256  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
```

```
3197  015260  010412                        MOV     R4,(R2)         ;MAINTENANCE REGISTER.
3198  015262  005012                ML1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
3199                                                                 ;INSTRUCTION SHOULD CAUSE
3200                                                                 ;A PARITY ERROR IN THE
3201                                                                 ;CACHE DATA MEMORY GROUP GP.
3202
3203
3204  015264                        ML2:                            ;REPORT ERROR.  MAINTENANCE
3205  015264  010437  001230                MOV     R4,$TMP2        ;FUNCTION FAILED TO
3206                                                                 ;CAUSE ERROR.
3207  015270  104127                1$:     ERROR   127
3208  015272  012737  177777  032334  MOV   #-1,MANFL2
3209  015300  000500                        BR      MLDONE
3210
3211  015302  022737  004500  177744  MLERR0: CMP     #4500,@#MEMERR  ;DID THE ERROR REGISTER
3212  015310  001042                        BNE     69$             ;SET PROPERLY?
3213
3214  015312  022626                64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3215  015314  005037  177572        65$:    CLR     @#MMR0
3216  015320  005037  172516                CLR     @#MMR3
3217  015324  012737  177777  177744        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3218  015332  005737  177744                TST     @#MEMERR        ;REGISTER.
3219  015336  001416                        BEQ     68$
3220
3221  015340                        66$:                            ;ERROR REGISTER WON'T
3222  015340  013737  177740  001230        MOV     @#LOADRS,$TMP2  ;CLEAR
3223  015346  013737  177742  001232        MOV     @#HIADRS,$TMP3
3224  015354  013737  177744  001234        MOV     @#MEMERR,$TMP4
3225
3226  015362  104130                67$:    ERROR   130
3227  015364  012737  177777  032314        MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3228  015372  000443                        BR      MLDONE
3229
3230  015374  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3231  015402  001356                        BNE     66$             ;UNLOCKED.
3232  015404  022737  000003  177742        CMP     #3,@#HIADRS
3233  015412  001352                        BNE     66$
3234  015414  000432                        BR      MLDONE
3235
3236  015416                        69$:                            ;REPORT ERROR REGISTER
3237  015416  012637  001230                MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3238  015422  005726                        TST     (SP)+           ;RESET THE STACK.
3239  015424  013737  177740  001232        MOV     @#LOADRS,$TMP3
3240  015432  013737  177742  001234        MOV     @#HIADRS,$TMP4
3241  015440  012737  000020  001236        MOV     #20,$TMP5
3242  015446  012737  004500  001240        MOV     #4500,$TMP6
3243  015454  013737  177744  001242        MOV     @#MEMERR,$TMP7
3244
3245  015462  104131                70$:    ERROR   131
3246  015464  012737  177777  032334        MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3247  015472  012737  177777  032330        MOV     #-1,MMRFL2
3248  015500  000705                        BR      65$
3249  015502  104416                MLDONE: RSET
3250
3251
3252                                ;;*****************************************************************
```

```
                                        ;*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.
                                        ;*
                                        ;;*********************************************************************
015504  000004                  TST31:  SCOPE
015506  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000031                  MN=$TN-1
                                                                ;SET THE SKAD REGISTER
015514  012737  016050  032100          MOV     #TST32,SKAD     ;IN CASE THE TEST ABORTS.

015522  113737  001102  001224          MOVB    $TSTNM,$TMP0

015530  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
015532  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
015534  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
015536  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
015540  012737  015646  000114          MOV     #NMERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
015546  012704  000040                  MOV     #40,R4  ;PATTERN TO BE PUT IN MAINT. REG.
015552  012702  177750                  MOV     #MAINT,R2
015556  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
                                                                ;FORCE MISS THE OTHER
                                                                ;GROUP
015564  012705  015626                  MOV     #NM1,R5         ;MAKE NM1 A HIT IN
015570  005715                          TST     (R5)            ;GROUP GP.
015572  005715                          TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
015574  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
015602  001007                          BNE     1$

                                                                ;IF NOT ERROR!
015604  010537  001230                  MOV     R5,$TMP2
015610  012737  000000  001226          MOV     #0,$TMP1
015616  104001                          ERROR   1

015620  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

                                1$:     NOP
015622  000240                          MOV     R4,(R2)         ;PUT THE PATTERN IN THE
015624  010412                                                  ;MAINTENANCE REGISTER.
015626  005012                  NM1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
                                                                ;CACHE DATA MEMORY GROUP GP.


                                NM2:                            ;REPORT ERROR.  MAINTENANCE
015630  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
015634  104127                  1$:     ERROR   127
015636  012737  177777  032334  MOV     #-1,MANFL2
015644  000500                          BR      NMDONE
```

```
3309
3310  015646  022737  004500  177744  NMERR0: CMP     #4500,@#MEMERR  ;DID THE ERROR REGISTER
3311  015654  001042                          BNE     69$             ;SET PROPERLY?
3312
3313  015656  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3314  015660  005037  177572          65$:    CLR     @#MMR0
3315  015664  005037  172516                  CLR     @#MMR3
3316  015670  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3317  015676  005737  177744                  TST     @#MEMERR        ;REGISTER.
3318  015702  001416                          BEQ     68$
3319
3320  015704                          66$:                            ;ERROR REGISTER WON'T
3321  015704  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
3322  015712  013737  177742  001232          MOV     @#HIADRS,$TMP3
3323  015720  013737  177744  001234          MOV     @#MEMERR,$TMP4
3324
3325  015726  104130                  67$:    ERROR   130
3326  015730  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3327  015736  000443                          BR      NMDONE
3328
3329  015740  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3330  015746  001356                          BNE     66$             ;UNLOCKED.
3331  015750  022737  000003  177742          CMP     #3,@#HIADRS
3332  015756  001352                          BNE     66$
3333  015760  000432                          BR      NMDONE
3334
3335  015762                          69$:                            ;REPORT ERROR REGISTER
3336  015762  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3337  015766  005726                          TST     (SP)+           ;RESET THE STACK.
3338  015770  013737  177740  001232          MOV     @#LOADRS,$TMP3
3339  015776  013737  177742  001234          MOV     @#HIADRS,$TMP4
3340  016004  012737  000040  001236          MOV     #40,$TMP5
3341  016012  012737  004500  001240          MOV     #4500,$TMP6
3342  016020  013737  177744  001242          MOV     @#MEMERR,$TMP7
3343
3344  016026  104131                  70$:    ERROR   131
3345  016030  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3346  016036  012737  177777  032330          MOV     #-1,MMRFL2
3347  016044  000705                          BR      65$
3348  016046  104416                  NMDONE: RSET
3349
3350
3351                                  ;;**********************************************************************
3352                                  ;*TEST 32        CACHE MAINTENANCE AND ERROR REGISTERS TEST 16
3353                                  ;*
3354                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3355                                  ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
3356                                  ;*LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3357                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3358                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3359                                  ;*TO THE CACHE.
3360                                  ;*
3361                                  ;;**********************************************************************
3362  016050  000004                  TST32:  SCOPE
3363  016052  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3364          000032                  MO=$TN-1
```

```
                                                        ;SET THE SKAD REGISTER
016060  012737  016414  032100          MOV     #TST33,SKAD     ;IN CASE THE TEST ABORTS.

016066  113737  001102  001224          MOVB    $TSTNM,$TMP0

016074  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
016076  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
016100  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
016102  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
016104  012737  016212  000114          MOV     #MOERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
016112  012704  000100                  MOV     #100,R4 ;PATTERN TO BE PUT IN MAINT. REG.
016116  012702  177750                  MOV     #MAINT,R2
016122  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
                                                        ;FORCE MISS THE OTHER
                                                        ;GROUP
016130  012705  016172                  MOV     #M01,R5         ;MAKE M01 A HIT IN
016134  005715                          TST     (R5)            ;GROUP GP.
016136  005715                          TST     (R5)

                                                        ;SEE IF REFERENCE ADDRESS
016140  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
016146  001007                          BNE     1$
                                                        ;IF NOT ERROR!
016150  010537  001230                  MOV     R5,$TMP2
016154  012737  000001  001226          MOV     #1,$TMP1
016162  104001                          ERROR   1

016164  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

016166  000240          1$:     NOP                     ;PUT THE PATTERN IN THE
016170  010412                  MOV     R4,(R2)         ;MAINTENANCE REGISTER.
016172  005012          M01:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                        ;INSTRUCTION SHOULD CAUSE
                                                        ;A PARITY ERROR IN THE
                                                        ;CACHE DATA MEMORY GROUP GP.


016174          M02:                            ;REPORT ERROR.  MAINTENANCE
016174  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                        ;CAUSE ERROR.
016200  104127          1$:     ERROR   127
016202  012737  177777  032334  MOV     #-1,MANFL2
016210  000500                  BR      MODONE

016212  022737  004600  177744  MOERR0: CMP     #4600,@#MEMERR  ;DID THE ERROR REGISTER
016220  001042                  BNE     69$             ;SET PROPERLY?

016222  022626          64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
016224  005037  177572  65$:    CLR     @#MMR0
016230  005037  172516          CLR     @#MMR3
016234  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
016242  005737  177744                  TST     @#MEMERR        ;REGISTER.
016246  001416                  BEQ     68$

016250          66$:                            ;ERROR REGISTER WON'T
016250  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
```

```
3421  016256  013737  177742  001232          MOV     @#HIADRS,$TMP3
3422  016264  013737  177744  001234          MOV     @#MEMERR,$TMP4
3423
3424  016272  104130                  67$:    ERROR   130
3425  016274  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3426  016302  000443                          BR      MODONE
3427
3428  016304  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3429  016312  001356                          BNE     66$             ;UNLOCKED.
3430  016314  022737  000003  177742          CMP     #3,@#HIADRS
3431  016322  001352                          BNE     66$
3432  016324  000432                          BR      MODONE
3433
3434  016326                          69$:                            ;REPORT ERROR REGISTER
3435  016326  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3436  016332  005726                          TST     (SP)+           ;RESET THE STACK.
3437  016334  013737  177740  001232          MOV     @#LOADRS,$TMP3
3438  016342  013737  177742  001234          MOV     @#HIADRS,$TMP4
3439  016350  012737  000100  001236          MOV     #100,$TMP5
3440  016356  012737  004600  001240          MOV     #4600,$TMP6
3441  016364  013737  177744  001242          MOV     @#MEMERR,$TMP7
3442
3443  016372  104131                  70$:    ERROR   131
3444  016374  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3445  016402  012737  177777  032330          MOV     #-1,MMRFL2
3446  016410  000705                          BR      65$
3447  016412  104416              MODONE: RSET
3448
3449
3450                                  ;;*****************************************************************
3451                                  ;*TEST 33       CACHE MAINTENANCE AND ERROR REGISTERS TEST 17
3452                                  ;*
3453                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3454                                  ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
3455                                  ;*HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3456                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3457                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3458                                  ;*TO THE CACHE.
3459                                  ;*
3460                                  ;;*****************************************************************
3461  016414  000004          TST33:  SCOPE
3462  016416  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3463          000033          MP=$TN-1
3464                                                                  ;SET THE SKAD REGISTER
3465  016424  012737  016760  032100          MOV     #TST34,SKAD     ;IN CASE THE TEST ABORTS.
3466
3467  016432  113737  001102  001224          MOVB    $TSTNM,$TMP0
3468
3469  016440  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3470  016442  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3471  016444  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3472  016446  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3473  016450  012737  016556  000114          MOV     #MPERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
3474  016456  012704  000200                  MOV     #200,R4 ;PATTERN TO BE PUT IN MAINT. REG.
3475  016462  012702  177750                  MOV     #MAINT,R2
3476  016466  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
```

```
                                                        ;FORCE MISS THE OTHER
                                                        ;GROUP
016474  012705  016536                MOV     #MP1,R5     ;MAKE MP1 A HIT IN
016500  005715                        TST     (R5)        ;GROUP GP.
016502  005715                        TST     (R5)
                                                        ;SEE IF REFERENCE ADDRESS
016504  032737  000010  177752        BIT     #10,@#HITMIS ;IS A HIT.
016512  001007                        BNE     1$
                                                        ;IF NOT ERROR!
016514  010537  001230                MOV     R5,$TMP2
016520  012737  000001  001226        MOV     #1,$TMP1
016526  104001                        ERROR   1
                                                        ;ERROR FATAL. GO TO NEXT TEST.
016530  104420                        SKIPT
                                                        ;PUT THE PATTERN IN THE
016532  000240                1$:     NOP
016534  010412                        MOV     R4,(R2)     ;MAINTENANCE REGISTER.
016536  005012                MP1:    CLR     (R2)        ;THE FETCH OF THIS NEXT
                                                        ;INSTRUCTION SHOULD CAUSE
                                                        ;A PARITY ERROR IN THE
                                                        ;CACHE DATA MEMORY GROUP GP.

016540                          MP2:
                                                        ;REPORT ERROR.  MAINTENANCE
016540  010437  001230                MOV     R4,$TMP2    ;FUNCTION FAILED TO
                                                        ;CAUSE ERROR.
016544  104127                  1$:   ERROR   127
016546  012737  177777  032334  MOV   #-1,MANFL2
016554  000500                        BR      MPDONE

016556  022737  004600  177744  MPERR0: CMP   #4600,@#MEMERR ;DID THE ERROR REGISTER
016564  001042                        BNE     69$         ;SET PROPERLY?

016566  022626                64$:    CMP     (SP)+,(SP)+ ;RESET THE STACK
016570  005037  177572        65$:    CLR     @#MMR0
016574  005037  172516                CLR     @#MMR3
016600  012737  177777  177744        MOV     #-1,@#MEMERR ;TRY TO CLEAR THE ERROR
016606  005737  177744                TST     @#MEMERR    ;REGISTER.
016612  001416                        BEQ     68$

016614                          66$:
                                                        ;ERROR REGISTER WON'T
016614  013737  177740  001230        MOV     @#LOADRS,$TMP2 ;CLEAR
016622  013737  177742  001232        MOV     @#HIADRS,$TMP3
016630  013737  177744  001234        MOV     @#MEMERR,$TMP4

016636  104130                67$:    ERROR   130
016640  012737  177777  032314        MOV     #-1,MMRFLG  ;SIGNAL BAD REGISTER
016646  000443                        BR      MPDONE

016650  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
016656  001356                        BNE     66$         ;UNLOCKED.
016660  022737  000003  177742        CMP     #3,@#HIADRS
016666  001352                        BNE     66$
016670  000432                        BR      MPDONE
```

```
3533  016672                                69$:                                    ;REPORT ERROR REGISTER
3534  016672  012637  001230                        MOV     (SP)+,$TMP2             ;NOT SET AS EXPECTED.
3535  016676  005726                                TST     (SP)+                   ;RESET THE STACK.
3536  016700  013737  177740  001232                MOV     @#LOADRS,$TMP3
3537  016706  013737  177742  001234                MOV     @#HIADRS,$TMP4
3538  016714  012737  000200  001236                MOV     #200,$TMP5
3539  016722  012737  004600  001240                MOV     #4600,$TMP6
3540  016730  013737  177744  001242                MOV     @#MEMERR,$TMP7
3541
3542  016736  104131                        70$:    ERROR   131
3543  016740  012737  177777  032334                MOV     #-1,MANFL2              ;SIGNAL BAD REGISTER
3544  016746  012737  177777  032330                MOV     #-1,MMRFL2
3545  016754  000705                                BR      65$
3546  016756  104416                        MPDONE: RSET
3547
3548
3549
3550
3551
3552                                  ;;**********************************************************************
3553                                  ;*TEST 34       CACHE MAINTENANCE AND ERROR REGISTERS TEST 20
3554                                  ;*
3555                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3556                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3557                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3558                                  ;*THE MAINTENANCE REGISTER IS USED TO MAKE THAT REFERENCE CAUSE A
3559                                  ;*MAIN MEMORY ADDRESS AND CONTROL LINES PARITY ERROR ON THE
3560                                  ;*MAIN MEMORY BUS.
3561                                  ;*
3562                                  ;;**********************************************************************
3563  016760  000004                  TST34:  SCOPE
3564  016762  012737  000040  001274          MOV     #40,$TIMES              ;;DO 40 ITERATIONS
3565          000034                  MR=$TN-1
3566                                                                          ;SET THE SKAD REGISTER
3567  016770  012737  017410  032100          MOV     #TST35,SKAD             ;IN CASE THE TEST ABORTS.
3568
3569  016776  113737  001102  001224          MOVB    $TSTNM,$TMP0
3570
3571  017004  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3572  017006  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3573  017010  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3574  017012  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3575  017014  104422                          MMSKIP
3576  017016  012737  017200  000114          MOV     #MRERR0,@#CACHVEC               ;SET UP FOR THE ERROR.
3577  017024  012737  031726  000004          MOV     #CPSPUR,@#ERRVEC                ;NOTE THAT WHEN THIS ERROR
3578                                                                                  ;ON THE MAIN MEMORY ADDRESS
3579                                                                                  ;AND CONTROL LINES OCCURS
3580                                                                                  ;A TIME OUT WILL RESULT ON THE
3581                                                                                  ;UNIBUS!! THIS WILL CAUSE A
3582                                                                                  ;TRAP TO VECTOR ERRVEC BEFORE
3583                                                                                  ;THE TRAP TO CACHVEC OCCURS! BOTH
3584                                                                                  ;WILL OCCUR!
3585  017032  012746  177777                  MOV     #-1,-(SP)                       ;PUT A MARKER ON THE STACK
3586
3587  017036  012700  172340                  MOV     #KIPAR0,R0              ;SET UP MEMORY MANAGEMENT
3588                                                                          ;TO RELOCATE EVERYTHING
```

```
3589  017042  012702  172300              MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
3590  017046  012703  000007              MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
3591  017052  005004                      CLR     R4              ;BY PASSIVELY IS MEANT
3592  017054  012705  170200              MOV     #MAPL00,R5      ;THAT ADDRESS ARE
3593                                                              ;RELOCATED TO THEMSELVES.
3594  017060  012722  077406      64$:    MOV     #77406,(R2)+
3595  017064  010401                      MOV     R4,R1
3596  017066  072127  000006              ASH     #6,R1
3597  017072  010125                      MOV     R1,(R5)+
3598  017074  005025                      CLR     (R5)+
3599  017076  010410                      MOV     R4,(R0)
3600  017100  062720  170000              ADD     #170000,(R0)+
3601  017104  062704  000200              ADD     #200,R4
3602  017110  077315                      SOB     R3,64$
3603  017112  012710  177600              MOV     #177600,(R0)
3604  017116  012712  077406              MOV     #77406,(R2)
3605
3606  017122  012737  000060  172516      MOV     #60,@#MMR3              ;TURN ON THE MAPPING BOX AND
3607  017130  012737  000001  177572      MOV     #1,@#MMR0               ;ENABLE 22 BIT MODE ADDRESSING.
3608
3609  017136  012737  000014  177746      MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
3610  017144  012702  177750              MOV     #MAINT,R2
3611  017150  000240                      NOP                             ;FOR SCOPING WITH AN OSCILLOSCOPE!
3612  017152  012712  000002              MOV     #2,(R2)                 ;SET UP THE FORCE ERROR BIT IN
3613                                                                      ;THE MAINTENANCE REGISTER.
3614  017156  005012                      CLR     (R2)                    ;THE FETCH OF THIS INSTRUCTION
3615                                                                      ;SHOULD RESULT IN A PARITY ERROR
3616                                                                      ;ON THE MAIN MEMORY ADDRESS AND CONTROL
3617                                                                      ;LINES. BECAUSE THIS REFERENCE
3618                                                                      ;IS BEING MADE OVER THE UNIBUS
3619                                                                      ;A UNIBUS TIME OUT WILL OCCUR
3620                                                                      ;RESULTING IN AN ABORT TO VECTOR
3621                                                                      ;ERRVEC. THEN IMMEDIATELY FOLLOWING
3622                                                                      ;THIS ABORT TO ERRVEC, THE
3623                                                                      ;PARITY ERROR WILL CAUSE A TRAP
3624                                                                      ;TO CACHVEC!!!
3625
3626  017160                      MR1:                                    ;REPORT FAILURE OF THE MAINTENANCE
3627  017160  012737  000002  001230      MOV     #2,$TMP2                ;TO FORCE THE ERROR.
3628  017166  104127              1$:     ERROR   127
3629  017170  012737  177777  032334      MOV     #-1,MANFL2
3630  017176  000503                      BR      MRDONE
3631
3632  017200  022766  177777  000010  MRERR0: CMP    #-1,10(SP)              ;DID 2 TRAPS OCCUR? SEE WHERE
3633                                                                      ;THE MARKER IS ON THE STACK!
3634  017206  001401                      BEQ     MR2
3635  017210  104000                      ERROR
3636
3637  017212  022737  002402  177744  MR2:    CMP     #2402,@#MEMERR          ;DID THE ERROR REGISTER GET
3638  017220  001430                      BEQ     MR3                     ;SET CORRECTLY.
3639
3640                                                                      ;IF NOT REPORT THE ERROR.
3641  017222  022626                      CMP     (SP)+,(SP)+
3642  017224  012637  001230              MOV     (SP)+,$TMP2
3643  017230  022626                      CMP     (SP)+,(SP)+
3644  017232  013737  177740  001232      MOV     @#LOADRS,$TMP3
```

```
3645  017240  013737  177742  001234          MOV     @#HIADRS,$TMP4
3646  017246  012737  000002  001236          MOV     #2,$TMP5
3647  017254  012737  002402  001240          MOV     #2402,$TMP6
3648  017262  013737  177744  001242          MOV     @#MEMERR,$TMP7
3649  017270  104131                  1$:     ERROR   131
3650  017272  012737  177777  032334          MOV     #-1,MANFL2
3651  017300  000402                          BR      MR4
3652
3653  017302  062706  000012          MR3:    ADD     #12,SP                  ;RESET THE STACK.
3654
3655  017306  005037  177572          MR4:    CLR     @#MMR0
3656  017312  005037  172516                  CLR     @#MMR3
3657  017316  012737  177777  177744          MOV     #-1,@#MEMERR            ;TRY TO CLR THE ERROR REG.
3658  017324  005737  177744                  TST     @#MEMERR
3659  017330  001416                          BEQ     MR6
3660
3661  017332                  MR5:                                            ;THE ERROR REGISTER WON'T CLR.
3662  017332  013737  177740  001230          MOV     @#LOADRS,$TMP2
3663  017340  013737  177742  001232          MOV     @#HIADRS,$TMP3
3664  017346  013737  177744  001234          MOV     @#MEMERR,$TMP4
3665  017354  104130                  1$:     ERROR   130
3666  017356  012737  177777  032314          MOV     #-1,MMRFLG
3667  017364  000410                          BR      MRDONE
3668
3669  017366  022737  177740  177740  MR6:    CMP     #177740,@#LOADRS        ;SEE IF THE ADDRESS REGISTER
3670  017374  001356                          BNE     MR5                     ;GOT RESET.
3671  017376  022737  000003  177742          CMP     #3,@#HIADRS
3672  017404  001352                          BNE     MR5
3673
3674  017406  104416          MRDONE: RSET
3675
3676                          ;;*******************************************************************
3677                          ;*TEST 35       CACHE MAINTENANCE AND ERROR REGISTERS TEST 21
3678                          ;*
3679                          ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3680                          ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3681                          ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3682                          ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
3683                          ;*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN EVEN WORD IN THE
3684                          ;*PAIR, WHICH IS ALSO THE WANTED WORD.
3685                          ;*
3686                          ;;*******************************************************************
3687  017410  000004          TST35:  SCOPE
3688  017412  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3689          000035          MS=$TN-1
3690                                                  ;SET THE SKAD REGISTER
3691  017420  012737  020030  032100          MOV     #TST36,SKAD     ;IN CASE THE TEST ABORTS.
3692
3693  017426  113737  001102  001224          MOVB    $TSTNM,$TMP0
3694
3695  017434  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3696  017436  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3697  017440  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3698  017442  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3699  017444  104422                          MMSKIP
3700  017446  012737  017626  000114          MOV     #MSERR0,@#CACHVEC       ;SET UP FOR THE ERROR
```

```
3701
3702  017454  012700  172340          MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
3703                                                          ;TO RELOCATE EVERYTHING
3704  017460  012702  172300          MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
3705  017464  012703  000007          MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
3706  017470  005004                  CLR     R4              ;BY PASSIVELY IS MEANT
3707  017472  012705  170200          MOV     #MAPL00,R5      ;THAT ADDRESS ARE
3708                                                          ;RELOCATED TO THEMSELVES.
3709  017476  012722  077406  64$:    MOV     #77406,(R2)+
3710  017502  010401                  MOV     R4,R1
3711  017504  072127  000006          ASH     #6,R1
3712  017510  010125                  MOV     R1,(R5)+
3713  017512  005025                  CLR     (R5)+
3714  017514  010410                  MOV     R4,(R0)
3715  017516  062720  170000          ADD     #170000,(R0)+
3716  017522  062704  000200          ADD     #200,R4
3717  017526  077315                  SOB     R3,64$
3718  017530  012710  177600          MOV     #177600,(R0)
3719  017534  012712  077406          MOV     #77406,(R2)
3720
3721  017540  012737  000060  172516  MOV     #60,@#MMR3              ;TURN THE MAP AND ENABLE
3722  017546  012737  000001  177572  MOV     #1,@#MMR0               ;22 BIT MODE ADDRESSING.
3723  017554  012704  010000          MOV     #10000,R4               ;PATTERN FOR THE MAINTENANCE
3724  017560  012702  177750          MOV     #MAINT,R2               ;REGISTER.
3725  017564  012737  000014  177746  MOV     #M1M0,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
3726  017572  000402                  BR      MS1
3727
3728          017574                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
3729          017574                  LOC=-4&LOC
3730          017600                  LOC=LOC+4
3731          017600                  .=LOC
3732
3733  017600  000240          MS1:    NOP
3734  017602  010412                  MOV     R4,(R2)                 ;TURN ON THE MAINTENANCE REGISTER.
3735  017604  005701          MS2:    TST     R1
3736  017606  005012                  CLR     (R2)
3737
3738  017610                  MS3:                                    ;REPORT ERROR.  MAINTENANCE
3739  017610  010437  001230          MOV     R4,$TMP2                ;FUNCTION FAILED TO
3740                                                                  ;CAUSE ERROR.
3741  017614  104127          1$:     ERROR   127
3742  017616  012737  177777  032334  MOV     #-1,MANFL2
3743  017624  000500                  BR      MSDONE
3744
3745  017626  022737  023404  177744  MSERR0: CMP     #23404,@#MEMERR ;DID THE ERROR REGISTER
3746  017634  001042                  BNE     69$                     ;SET PROPERLY?
3747
3748  017636  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3749  017640  005037  177572          65$:    CLR     @#MMR0
3750  017644  005037  172516                  CLR     @#MMR3
3751  017650  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3752  017656  005737  177744                  TST     @#MEMERR        ;REGISTER.
3753  017662  001416                  BEQ     68$
3754
3755  017664                  66$:                                    ;ERROR REGISTER WON'T
3756  017664  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
```

```
3757  017672  013737  177742  001232          MOV     @#HIADRS,$TMP3
3758  017700  013737  177744  001234          MOV     @#MEMERR,$TMP4
3759
3760  017706  104130                  67$:    ERROR   130
3761  017710  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3762  017716  000443                          BR      MSDONE
3763
3764  017720  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3765  017726  001356                          BNE     66$             ;UNLOCKED.
3766  017730  022737  000003  177742          CMP     #3,@#HIADRS
3767  017736  001352                          BNE     66$
3768  017740  000432                          BR      MSDONE
3769
3770  017742                          69$:                            ;REPORT ERROR REGISTER
3771  017742  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3772  017746  005726                          TST     (SP)+           ;RESET THE STACK.
3773  017750  013737  177740  001232          MOV     @#LOADRS,$TMP3
3774  017756  013737  177742  001234          MOV     @#HIADRS,$TMP4
3775  017764  012737  010000  001236          MOV     #10000,$TMP5
3776  017772  012737  023404  001240          MOV     #23404,$TMP6
3777  020000  013737  177744  001242          MOV     @#MEMERR,$TMP7
3778
3779  020006  104131                  70$:    ERROR   131
3780  020010  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3781  020016  012737  177777  032330          MOV     #-1,MMRFL2
3782  020024  000705                          BR      65$
3783  020026  104416                  MSDONE: RSET
3784
3785                                  ;;*****************************************************************
3786                                  ;*TEST 36       CACHE MAINTENANCE AND ERROR REGISTERS TEST 22
3787                                  ;*
3788                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3789                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3790                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3791                                  ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
3792                                  ;*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN ODD WORD IN THE
3793                                  ;*PAIR, WHICH IS ALSO THE WANTED WORD.
3794                                  ;*
3795                                  ;;*****************************************************************
3796  020030  000004                  TST36:  SCOPE
3797  020032  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3798          000036          MT=$TN-1
3799                                                                  ;SET THE SKAD REGISTER
3800  020040  012737  020454  032100          MOV     #TST37,SKAD     ;IN CASE THE TEST ABORTS.
3801
3802  020046  113737  001102  001224          MOVB    $TSTNM,$TMP0
3803
3804  020054  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3805  020056  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3806  020060  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3807  020062  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3808  020064  104422                          MMSKIP
3809
3810  020066  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
3811                                                                  ;TO RELOCATE EVERYTHING
3812  020072  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
```

```
3813  020076  012703  000007                MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
3814  020102  005004                       CLR     R4              ;BY PASSIVELY IS MEANT
3815  020104  012705  170200               MOV     #MAPL00,R5      ;THAT ADDRESS ARE
3816                                                                 ;RELOCATED TO THEMSELVES.
3817  020110  012722  077406        64$:   MOV     #77406,(R2)+
3818  020114  010401                       MOV     R4,R1
3819  020116  072127  000006               ASH     #6,R1
3820  020122  010125                       MOV     R1,(R5)+
3821  020124  005025                       CLR     (R5)+
3822  020126  010410                       MOV     R4,(R0)
3823  020130  062720  170000               ADD     #170000,(R0)+
3824  020134  062704  000200               ADD     #200,R4
3825  020140  077315                       SOB     R3,64$
3826  020142  012710  177600               MOV     #177600,(R0)
3827  020146  012712  077406               MOV     #77406,(R2)
3828
3829  020152  012737  000060  172516       MOV     #60,@#MMR3      ;TURN ON THE MAP AND 22-BIT
3830  020160  012737  000001  177572       MOV     #1,@#MMR0       ;MODE ADDRESSING.
3831  020166  012737  020252  000114       MOV     #MTERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
3832  020174  012737  000014  177746       MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
3833  020202  012704  040000               MOV     #40000,R4       ;PATTERN TO BE PUT IN MAINT.
3834  020206  012702  177750               MOV     #MAINT,R2       ;REG.
3835  020212  000403                       BR      MT1
3836
3837          020214                       LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
3838          020214                       LOC=-4&LOC
3839          020220                       LOC=LOC+4
3840          020220                       .=LOC
3841
3842  020220  000240                       NOP
3843  020222  000240               MT1:    NOP                     ;NOP FOR SCOPING WITH AN OSCILLOSCOPE!!
3844  020224  010412                       MOV     R4,(R2)         ;SET THE MAINT. REG.
3845  020226  005701                       TST     R1              ;THE REFERENCE TO THIS INSTRUCTION SHOULD CAUSE A PARITY
3846  020230  005012                       CLR     (R2)            ;ABORT CAUSED BY DETECTION OF BAD PARITY ON
3847  020232  000240                       NOP                     ;THE WANTED, ODD, WORD IN THIS PAIR.
3848
3849
3850  020234                       MT2:                            ;REPORT ERROR.  MAINTENANCE
3851  020234  010437  001230               MOV     R4,$TMP2        ;FUNCTION FAILED TO
3852                                                                 ;CAUSE ERROR.
3853  020240  104127               1$:     ERROR   127
3854  020242  012737  177777  032334 MOV    #-1,MANFL2
3855  020250  000500                       BR      MTDONE
3856
3857  020252  022737  023410  177744 MTERR0: CMP    #23410,@#MEMERR ;DID THE ERROR REGISTER
3858  020260  001042                       BNE     69$             ;SET PROPERLY?
3859
3860  020262  022626                64$:   CMP     (SP)+,(SP)+     ;RESET THE STACK
3861  020264  005037  177572        65$:   CLR     @#MMR0
3862  020270  005037  172516               CLR     @#MMR3
3863  020274  012737  177777  177744       MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3864  020302  005737  177744               TST     @#MEMERR        ;REGISTER.
3865  020306  001416                       BEQ     68$
3866
3867  020310                       66$:                            ;ERROR REGISTER WON'T
3868  020310  013737  177740  001230       MOV     @#LOADRS,$TMP2  ;CLEAR
```

```
3869  020316  013737  177742  001232          MOV     @#HIADRS,$TMP3
3870  020324  013737  177744  001234          MOV     @#MEMERR,$TMP4
3871
3872  020332  104130                  67$:    ERROR   130
3873  020334  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3874  020342  000443                          BR      MTDONE
3875
3876  020344  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3877  020352  001356                          BNE     66$             ;UNLOCKED.
3878  020354  022737  000003  177742          CMP     #3,@#HIADRS
3879  020362  001352                          BNE     66$
3880  020364  000432                          BR      MTDONE
3881
3882  020366                          69$:                            ;REPORT ERROR REGISTER
3883  020366  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3884  020372  005726                          TST     (SP)+           ;RESET THE STACK.
3885  020374  013737  177740  001232          MOV     @#LOADRS,$TMP3
3886  020402  013737  177742  001234          MOV     @#HIADRS,$TMP4
3887  020410  012737  040000  001236          MOV     #40000,$TMP5
3888  020416  012737  023410  001240          MOV     #23410,$TMP6
3889  020424  013737  177744  001242          MOV     @#MEMERR,$TMP7
3890
3891  020432  104131                  70$:    ERROR   131
3892  020434  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3893  020442  012737  177777  032330          MOV     #-1,MMRFL2
3894  020450  000705                          BR      65$
3895  020452  104416                  MTDONE: RSET
3896
3897                                  ;;*****************************************************************
3898                                  ;*TEST 37        CACHE MAINTENANCE AND ERROR REGISTERS TEST 23
3899                                  ;*
3900                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3901                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3902                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3903                                  ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY
3904                                  ;*PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE
3905                                  ;*LOW BYTE OF THAT ADDRESS .
3906                                  ;*
3907                                  ;;*****************************************************************
3908  020454  000004          TST37:  SCOPE
3909  020456  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3910          000037          MU=$TN-1
3911                                                                  ;SET THE SKAD REGISTER
3912  020464  012737  021074  032100          MOV     #TST40,SKAD     ;IN CASE THE TEST ABORTS.
3913
3914  020472  113737  001102  001224          MOVB    $TSTNM,$TMP0
3915
3916  020500  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3917  020502  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3918  020504  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3919  020506  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3920  020510  104422                          MMSKIP
3921
3922  020512  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
3923                                                                  ;TO RELOCATE EVERYTHING
3924  020516  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
```

```
3925  020522  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
3926  020526  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
3927  020530  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
3928                                                                  ;RELOCATED TO THEMSELVES.
3929  020534  012722  077406          64$:    MOV     #77406,(R2)+
3930  020540  010401                          MOV     R4,R1
3931  020542  072127  000006                  ASH     #6,R1
3932  020546  010125                          MOV     R1,(R5)+
3933  020550  005025                          CLR     (R5)+
3934  020552  010410                          MOV     R4,(R0)
3935  020554  062720  170000                  ADD     #170000,(R0)+
3936  020560  062704  000200                  ADD     #200,R4
3937  020564  077315                          SOB     R3,64$
3938  020566  012710  177600                  MOV     #177600,(R0)
3939  020572  012712  077406                  MOV     #77406,(R2)
3940
3941  020576  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
3942  020604  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
3943  020612  012737  020672  000114          MOV     #MUERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
3944  020620  012737  000030  177746          MOV     #S0M1,@#CONTRL  ;SELECT GROUP ADDRESS
3945  020626  012704  000400                  MOV     #400,R4 ;PATTERN TO BE LOADED IN THE
3946  020632  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REG.
3947  020636  000403                          BR      MU1
3948
3949          020640                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
3950          020640                          LOC=-4&LOC
3951          020644                          LOC=LOC+4
3952          020644                          .=LOC
3953
3954  020644  000240                          NOP
3955  020646  000240                  .       NOP
3956  020650  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
3957  020652  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
3958                                                                  ;A PARITY ERROR IN GROUP
3959                                                                  ;ADDRESS 0 MEMORY
3960
3961  020654                          MU2:                            ;REPORT ERROR.  MAINTENANCE
3962  020654  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
3963                                                                  ;CAUSE ERROR.
3964  020660  104127                  1$:     ERROR   127
3965  020662  012737  177777  032334  MOV     #-1,MANFL2
3966  020670  000500                          BR      MUDONE
3967
3968  020672  022737  002420  177744  MUERR0: CMP     #2420,@#MEMERR  ;DID THE ERROR REGISTER
3969  020700  001042                          BNE     69$             ;SET PROPERLY?
3970
3971  020702  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3972  020704  005037  177572          65$:    CLR     @#MMR0
3973  020710  005037  172516                  CLR     @#MMR3
3974  020714  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3975  020722  005737  177744                  TST     @#MEMERR        ;REGISTER.
3976  020726  001416                          BEQ     68$
3977
3978  020730                          66$:                            ;ERROR REGISTER WON'T
3979  020730  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
3980  020736  013737  177742  001232          MOV     @#HIADRS,$TMP3
```

```
                                                MOV     @#MEMERR,$TMP4

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

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

                                69$:                                    ;REPORT ERROR REGISTER
                                                MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
                                                TST     (SP)+           ;RESET THE STACK.
                                                MOV     @#LOADRS,$TMP3
                                                MOV     @#HIADRS,$TMP4
                                                MOV     #400,$TMP5
                                                MOV     #2420,$TMP6
                                                MOV     @#MEMERR,$TMP7

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

                                ;;*****************************************************************
                                ;*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 .
                                ;*
                                ;;*****************************************************************
                                TST40:  SCOPE
                                                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                MV=$TN-1
                                                                        ;SET THE SKAD REGISTER
                                        MOV     #TST41,SKAD             ;IN CASE THE TEST ABORTS.

                                        MOVB    $TSTNM,$TMP0

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

                                        MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                ;TO RELOCATE EVERYTHING
                                        MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
                                        MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
```

```
4037  021146  005004                          CLR     R4                      ;BY PASSIVELY IS MEANT
4038  021150  012705  170200                  MOV     #MAPL00,R5              ;THAT ADDRESS ARE
4039                                                                          ;RELOCATED TO THEMSELVES.
4040  021154  012722  077406          64$:    MOV     #77406,(R2)+
4041  021160  010401                          MOV     R4,R1
4042  021162  072127  000006                  ASH     #6,R1
4043  021166  010125                          MOV     R1,(R5)+
4044  021170  005025                          CLR     (R5)+
4045  021172  010410                          MOV     R4,(R0)
4046  021174  062720  170000                  ADD     #170000,(R0)+
4047  021200  062704  000200                  ADD     #200,R4
4048  021204  077315                          SOB     R3,64$
4049  021206  012710  177600                  MOV     #177600,(R0)
4050  021212  012712  077406                  MOV     #77406,(R2)
4051
4052  021216  012737  000060  172516          MOV     #60,@#MMR3              ;TURN ON THE MAP AND
4053  021224  012737  000001  177572          MOV     #1,@#MMR0               ;22-BIT MODE ADDRESSING
4054  021232  012737  021312  000114          MOV     #MVERR0,@#CACHVEC               ;SETUP FOR THE ERROR.
4055  021240  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;SELECT GROUP ADDRESS
4056  021246  012704  002000                  MOV     #2000,R4                ;PATTERN TO BE LOADED IN THE
4057  021252  012702  177750                  MOV     #MAINT,R2               ;MAINTENANCE REG.
4058  021256  000403                          BR      MV1
4059
4060          021260                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4061          021260                          LOC=-4&LOC
4062          021264                          LOC=LOC+4
4063          021264                          .=LOC
4064
4065  021264  000240                          NOP
4066  021266  000240          MV1:    NOP
4067  021270  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
4068  021272  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
4069                                                                  ;A PARITY ERROR IN GROUP
4070                                                                  ;ADDRESS 1 MEMORY
4071
4072  021274                  MV2:                                    ;REPORT ERROR.  MAINTENANCE
4073  021274  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
4074                                                                  ;CAUSE ERROR.
4075  021300  104127          1$:     ERROR   127
4076  021302  012737  177777  032334  MOV     #-1,MANFL2
4077  021310  000500                          BR      MVDONE
4078
4079  021312  022737  002440  177744  MVERR0: CMP     #2440,@#MEMERR  ;DID THE ERROR REGISTER
4080  021320  001042                          BNE     69$             ;SET PROPERLY?
4081
4082  021322  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
4083  021324  005037  177572          65$:    CLR     @#MMR0
4084  021330  005037  172516                  CLR     @#MMR3
4085  021334  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
4086  021342  005737  177744                  TST     @#MEMERR        ;REGISTER.
4087  021346  001416                          BEQ     68$
4088
4089  021350                  66$:                                    ;ERROR REGISTER WON'T
4090  021350  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
4091  021356  013737  177742  001232          MOV     @#HIADRS,$TMP3
4092  021364  013737  177744  001234          MOV     @#MEMERR,$TMP4
```

```
                                67$:    ERROR   130
021372  104130                          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
021374  012737  177777  032314        BR      MVDONE
021402  000443
                                68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
021404  022737  177740  177740        BNE     66$             ;UNLOCKED.
021412  001356                        CMP     #3,@#HIADRS
021414  022737  000003  177742        BNE     66$
021422  001352                        BR      MVDONE
021424  000432
                                69$:                            ;REPORT ERROR REGISTER
021426                          MOV     (SP)+,$TMP2             ;NOT SET AS EXPECTED.
021426  012637  001230                TST     (SP)+               ;RESET THE STACK.
021432  005726                        MOV     @#LOADRS,$TMP3
021434  013737  177740  001232        MOV     @#HIADRS,$TMP4
021442  013737  177742  001234        MOV     #2000,$TMP5
021450  012737  002000  001236        MOV     #2440,$TMP6
021456  012737  002440  001240        MOV     @#MEMERR,$TMP7
021464  013737  177744  001242
                                70$:    ERROR   131
021472  104131                          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
021474  012737  177777  032334        MOV     #-1,MMRFL2
021502  012737  177777  032330        BR      65$
021510  000705
021512  104416                  MVDONE: RSET
                        ;;**********************************************************************
                        ;*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 .
                        ;*
                        ;;**********************************************************************
021514  000004         TST41:  SCOPE
021516  012737  000040  001274        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000041          MW=$TN-1
                                                            ;SET THE SKAD REGISTER
021524  012737  022134  032100        MOV     #TST42,SKAD     ;IN CASE THE TEST ABORTS.

021532  113737  001102  001224        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.
021550  104422                  MMSKIP

021552  012700  172340                MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                            ;TO RELOCATE EVERYTHING
021556  012702  172300                MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
021562  012703  000007                MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
021566  005004                        CLR     R4              ;BY PASSIVELY IS MEANT
```

```
4149  021570  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
4150                                                                    ;RELOCATED TO THEMSELVES.
4151  021574  012722  077406          64$:    MOV     #77406,(R2)+
4152  021600  010401                          MOV     R4,R1
4153  021602  072127  000006                  ASH     #6,R1
4154  021606  010125                          MOV     R1,(R5)+
4155  021610  005025                          CLR     (R5)+
4156  021612  010410                          MOV     R4,(R0)
4157  021614  062720  170000                  ADD     #170000,(R0)+
4158  021620  062704  000200                  ADD     #200,R4
4159  021624  077315                          SOB     R3,64$
4160  021626  012710  177600                  MOV     #177600,(R0)
4161  021632  012712  077406                  MOV     #77406,(R2)
4162
4163  021636  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
4164  021644  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
4165  021652  012737  021732  000114          MOV     #MWERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
4166  021660  012737  000030  177746          MOV     #S0M1,@#CONTRL  ;SELECT GROUP DATA
4167  021666  012704  000020                  MOV     #20,R4  ;PATTERN TO BE LOADED IN THE
4168  021672  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REG.
4169  021676  000403                          BR      MW1
4170
4171          021700                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4172          021700                          LOC=-4&LOC
4173          021704                          LOC=LOC+4
4174          021704                          .=LOC
4175
4176  021704  000240                          NOP
4177  021706  000240          MW1:            NOP
4178  021710  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
4179  021712  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
4180                                                                  ;A PARITY ERROR IN GROUP
4181                                                                  ;DATA 0 MEMORY
4182
4183  021714                  MW2:                                    ;REPORT ERROR.  MAINTENANCE
4184  021714  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
4185                                                                  ;CAUSE ERROR.
4186  021720  104127          1$:     ERROR   127
4187  021722  012737  177777  032334  MOV     #-1,MANFL2
4188  021730  000500                  BR      MWDONE
4189
4190  021732  022737  002500  177744  MWERR0: CMP     #2500,@#MEMERR  ;DID THE ERROR REGISTER
4191  021740  001042                          BNE     69$             ;SET PROPERLY?
4192
4193  021742  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
4194  021744  005037  177572          65$:    CLR     @#MMR0
4195  021750  005037  172516                  CLR     @#MMR3
4196  021754  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
4197  021762  005737  177744                  TST     @#MEMERR        ;REGISTER.
4198  021766  001416                          BEQ     68$
4199
4200  021770                  66$:                                    ;ERROR REGISTER WON'T
4201  021770  013737  177740  001230          MOV     @#LOADRS,$TMP2  ;CLEAR
4202  021776  013737  177742  001232          MOV     @#HIADRS,$TMP3
4203  022004  013737  177744  001234          MOV     @#MEMERR,$TMP4
4204
```

```
4205  022012  104130                    67$:    ERROR   130
4206  022014  012737  177777  032314            MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
4207  022022  000443                            BR      MWDONE
4208
4209  022024  022737  177740  177740    68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
4210  022032  001356                            BNE     66$             ;UNLOCKED.
4211  022034  022737  000003  177742            CMP     #3,@#HIADRS
4212  022042  001352                            BNE     66$
4213  022044  000432                            BR      MWDONE
4214
4215  022046                            69$:                            ;REPORT ERROR REGISTER
4216  022046  012637  001230                    MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
4217  022052  005726                            TST     (SP)+           ;RESET THE STACK.
4218  022054  013737  177740  001232            MOV     @#LOADRS,$TMP3
4219  022062  013737  177742  001234            MOV     @#HIADRS,$TMP4
4220  022070  012737  000020  001236            MOV     #20,$TMP5
4221  022076  012737  002500  001240            MOV     #2500,$TMP6
4222  022104  013737  177744  001242            MOV     @#MEMERR,$TMP7
4223
4224  022112  104131                    70$:    ERROR   131
4225  022114  012737  177777  032334            MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
4226  022122  012737  177777  032330            MOV     #-1,MMRFL2
4227  022130  000705                            BR      65$
4228  022132  104416                    MWDONE: RSET
4229
4230                                    ;;**********************************************************************
4231                                    ;*TEST 42        CACHE MAINTENANCE AND ERROR REGISTERS TEST 26
4232                                    ;*
4233                                    ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
4234                                    ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
4235                                    ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
4236                                    ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY
4237                                    ;*PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE
4238                                    ;*LOW BYTE OF THAT DATA .
4239                                    ;*
4240                                    ;;**********************************************************************
4241  022134  000004                    TST42:  SCOPE
4242  022136  012737  000040  001274            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4243          000042                    MX=$TN-1
4244                                                                    ;SET THE SKAD REGISTER
4245  022144  012737  022554  032100            MOV     #TST43,SKAD     ;IN CASE THE TEST ABORTS.
4246
4247  022152  113737  001102  001224            MOVB    $TSTNM,$TMP0
4248
4249  022160  104430                            SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4250  022162  104432                            SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4251  022164  104434                            SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4252  022166  104436                            SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4253  022170  104422                            MMSKIP
4254
4255  022172  012700  172340                    MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
4256                                                                    ;TO RELOCATE EVERYTHING
4257  022176  012702  172300                    MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
4258  022202  012703  000007                    MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
4259  022206  005004                            CLR     R4              ;BY PASSIVELY IS MEANT
4260  022210  012705  170200                    MOV     #MAPL00,R5      ;THAT ADDRESS ARE
```

```
                                                        ;RELOCATED TO THEMSELVES.
022214  012722  077406          64$:    MOV     #77406,(R2)+
022220  010401                          MOV     R4,R1
022222  072127  000006                  ASH     #6,R1
022226  010125                          MOV     R1,(R5)+
022230  005025                          CLR     (R5)+
022232  010410                          MOV     R4,(R0)
022234  062720  170000                  ADD     #170000,(R0)+
022240  062704  000200                  ADD     #200,R4
022244  077315                          SOB     R3,64$
022246  012710  177600                  MOV     #177600,(R0)
022252  012712  077406                  MOV     #77406,(R2)

022256  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
022264  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
022272  012737  022352  000114          MOV     #MXERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
022300  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;SELECT GROUP DATA
022306  012704  000100                  MOV     #100,R4 ;PATTERN TO BE LOADED IN THE
022312  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REG.
022316  000403                          BR      MX1

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

022324  000240                          NOP
022326  000240          MX1:            NOP
022330  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
022332  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
                                                                ;A PARITY ERROR IN GROUP
                                                                ;DATA 1 MEMORY

022334          MX2:                                            ;REPORT ERROR.  MAINTENANCE
022334  010437  001230                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
022340  104127          1$:     ERROR   127
022342  012737  177777  032334  MOV     #-1,MANFL2
022350  000500                  BR      MXDONE

022352  022737  002600  177744  MXERR0: CMP     #2600,@#MEMERR  ;DID THE ERROR REGISTER
022360  001042                  BNE     69$             ;SET PROPERLY?

022362  022626          64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
022364  005037  177572  65$:    CLR     @#MMR0
022370  005037  172516          CLR     @#MMR3
022374  012737  177777  177744  MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
022402  005737  177744          TST     @#MEMERR        ;REGISTER.
022406  001416                  BEQ     68$

022410          66$:                                    ;ERROR REGISTER WON'T
022410  013737  177740  001230  MOV     @#LOADRS,$TMP2  ;CLEAR
022416  013737  177742  001232  MOV     @#HIADRS,$TMP3
022424  013737  177744  001234  MOV     @#MEMERR,$TMP4

022432  104130          67$:    ERROR   130
```

```
4317  022434  012737  177777  032314          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
4318  022442  000443                          BR      MXDONE
4319
4320  022444  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
4321  022452  001356                          BNE     66$             ;UNLOCKED.
4322  022454  022737  000003  177742          CMP     #3,@#HIADRS
4323  022462  001352                          BNE     66$
4324  022464  000432                          BR      MXDONE
4325
4326  022466                          69$:                            ;REPORT ERROR REGISTER
4327  022466  012637  001230                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
4328  022472  005726                          TST     (SP)+           ;RESET THE STACK.
4329  022474  013737  177740  001232          MOV     @#LOADRS,$TMP3
4330  022502  013737  177742  001234          MOV     @#HIADRS,$TMP4
4331  022510  012737  000100  001236          MOV     #100,$TMP5
4332  022516  012737  002600  001240          MOV     #2600,$TMP6
4333  022524  013737  177744  001242          MOV     @#MEMERR,$TMP7
4334
4335  022532  104131                  70$:    ERROR   131
4336  022534  012737  177777  032334          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
4337  022542  012737  177777  032330          MOV     #-1,MMRFL2
4338  022550  000705                          BR      65$
4339  022552  104416                  MXDONE: RSET
4340
4341                                  ;;**********************************************************************
4342                                  ;*TEST 43        CACHE ERROR REGISTER UNIBUS TIME OUT TEST
4343                                  ;*
4344                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO COMPREHEND A
4345                                  ;*CPU TO UNIBUS THROUGH THE MAP TO THE CACHE REFERENCE WHICH
4346                                  ;*TIMES OUT IN MAIN MEMORY. MANY SUCH NON-EXISTENT MEMORY LOCATIONS
4347                                  ;*ARE CONVIENTLY GUARENTEED TO EXIST! ALL THE ADDRESSES
4348                                  ;*FROM 17000000 THROUGH 17777776 ARE ADDRESSES
4349                                  ;*WHICH CAN NOT EXIST. HERE ONLY ONE OF THESE ADDRESSES, 17777776,
4350                                  ;*WILL BE USED TO CAUSE A TIME OUT ON THE UNIBUS AN THE CONSEQUENT
4351                                  ;*ABORT TO VECTOR ERRVEC.
4352                                  ;*
4353                                  ;*NOTE: NEW MEMORY OPTIONS MAKE 2048K OF MEMORY A POSSIBILITY.
4354                                  ;*IF SIZELO REG. INDICATES THE PRESENCE OF MORE THAN 1920K MEMORY,
4355                                  ;*THIS TEST WILL BE MODIFIED SO THAT MEMORY MANAGEMENT ATTEMPTS TO
4356                                  ;*ACCESS ADDRESS 17760000. THE UNIBUS MAP WILL NOT RESPOND TO THIS
4357                                  ;*ADDRESS (NOR SHOULD ANY UNIBUS DEVICE) THUS GENERATING A UNIBUS
4358                                  ;*TIMEOUT.  (REV D0)
4359                                  ;*
4360                                  ;;**********************************************************************
4361  022554  000004          TST43:  SCOPE
4362  022556  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4363          000043          MQ=$TN-1
4364                                                                  ;SET THE SKAD REGISTER
4365  022564  012737  023224  032100          MOV     #TST44,SKAD     ;IN CASE THE TEST ABORTS.
4366
4367  022572  113737  001102  001224          MOVB    $TSTNM,$TMP0
4368  022600  012737  031754  000114          MOV     #SPUR,@#CACHVEC         ;EXPECT NO PARITY ERRORS.
4369
4370  022606  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4371  022610  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4372  022612  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
```

```
                                        SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
                                        MMSKIP
                                        MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
                                        MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
                                        MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
                                        MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
                                64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
                                        SOB     R3,64$          ;ADDRESSES.
                                        CLR     (R0)+
                                        MOV     #200,(R0)+
                                        MOV     #400,(R0)+
                                        MOV     #600,(R0)+
                                        MOV     #1000,(R0)+
                                        MOV     #1200,(R0)+
                                        MOV     #1400,(R0)+
                                        MOV     #177600,(R0)

                                        MOV     #60,@#MMR3              ;TURN ON THE MAPPING BOX
                                        MOV     #1,@#MMR0               ;AND 22 BIT MODE ADDRESSING.
                                        CMP     #167777,@#$LSTBK        ;IS THERE MORE THAN 1920K?
                                        BGE     1$                      ;BRANCH IF NOT
                                        MOV     #177600,@#MQVAR         ;ELSE MODIFY VALUE FOR KIPAR6
                                1$:     MOV     @#MQVAR,@#KIPAR6        ;MAKE KIPAR6 RELOCATE
                                                                        ;TO THE UNIBUS.
                                        MOV     #MQERR,@#ERRVEC         ;SET UP THE TIME OUT VECTOR.

                                        MOV     #-2,@#MAPL00            ;SET THE MAP REGISTER 0
                                        MOV     #77,@#MAPH00
                                        MOV     #140000,R0              ;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!
                                        NOP
                                        TST     (R0)                    ;MAKE THE REFERENCE!

                                MQ1:                                    ;NO TIME OUT OCCURRED, REPORT
                                        MOV     #-2,$TMP2               ;THE ERROR.
                                        MOV     #77,$TMP3
                                1$:     ERROR   132
                                        BR      MQDONE

                                MQVAR:  .WORD   170000          ;VALUE TO BE PUT INTO KIPAR6

                                MQERR:  BIT     #20,@#CPUERR            ;SEE IF A TIME OUT HAS CAUSED
                                        BNE     MQ2                     ;AN ABORT TO THIS ROUTINE.
                                        JMP     CPSPUR                  ;IF NOT GO TO THE SPURIOUS
                                                                        ;UNEXPECTED, CPU ERROR HANDLER.
                                MQ2:    CMP     #0,@#MEMERR             ;OTHERWISE SEE IF THE ERROR
                                        BEQ     MQ3                     ;REGISTER GOT SET CORRECTLY.

                                                                ;IF IT IS NOT SET CORRECTLY REPORT ERROR.
```

```
4429  023042  012637  001230              MOV     (SP)+,$TMP2
4430  023046  005726                      TST     (SP)+
4431  023050  013737  177740  001232      MOV     @#LOADRS,$TMP3
4432  023056  013737  177742  001234      MOV     @#HIADRS,$TMP4
4433  023064  012737  177776  001236      MOV     #-2,$TMP5
4434  023072  012737  000077  001240      MOV     #77,$TMP6
4435  023100  013737  177744  001242      MOV     @#MEMERR,$TMP7
4436  023106  104133              1$:     ERROR   133
4437  023110  012737  177777  032330      MOV     #-1,MMRFL2
4438  023116  000401                      BR      MQ4
4439
4440  023120  022626              MQ3:    CMP     (SP)+,(SP)+             ;RESET THE STACK
4441
4442  023122  005037  177572      MQ4:    CLR     @#MMR0
4443  023126  005037  172516              CLR     @#MMR3
4444  023132  012737  177777  177744      MOV     #-1,@#MEMERR            ;TRY TO CLEAR THE ERROR REGISTER.
4445  023140  005737  177744              TST     @#MEMERR
4446  023144  001416                      BEQ     MQ6
4447
4448  023146              MQ5:                                    ;REPORT THE FAILURE OF THE ERROR
4449  023146  013737  177740  001230      MOV     @#LOADRS,$TMP2          ;REGISTER TO CLEAR!
4450  023154  013737  177742  001232      MOV     @#HIADRS,$TMP3
4451  023162  013737  177744  001234      MOV     @#MEMERR,$TMP4
4452  023170  104130              1$:     ERROR   130
4453  023172  012737  177777  032314      MOV     #-1,MMRFLG
4454  023200  000410                      BR      MQDONE
4455
4456  023202  022737  177740  177740  MQ6:    CMP     #177740,@#LOADRS        ;SEE IF THE ADDRESS REGISTER
4457  023210  001356                      BNE     MQ5                     ;GOT RESET.
4458  023212  022737  000003  177742      CMP     #3,@#HIADRS
4459  023220  001352                      BNE     MQ5
4460
4461  023222  104416              MQDONE: RSET
4462
4463                                      ;;*********************************************************************
4464                                      ;*TEST 44       CACHE CONTROL REGISTER DISABLE TRAPS TEST 1
4465                                      ;*
4466                                      ;*THIS IS A TEST OF THE CONTROL REGISTER'S ABILITY TO DISABLE A TRAP
4467                                      ;*OCCURRING AS THE RESULT OF A MAIN MEMORY DATA PARITY ERROR IN THE
4468                                      ;*UNWANTED WORD OF THE REFERENCED PAIR. THE MAINTENANCE REGISTER IS
4469                                      ;*USED TO FORCE AN ERROR ON THE LOW BYTE OF THE ODD WORD WHEN REFERENCING
4470                                      ;*THE EVEN WORD OF THAT PAIR.
4471                                      ;*
4472                                      ;;*********************************************************************
4473  023224  000004              TST44:  SCOPE
4474  023226  012737  000040  001274      MOV     #40,$TIMES              ;;DO 40 ITERATIONS
4475          000044              KV=$TN-1
4476                                                                      ;SET THE SKAD REGISTER
4477  023234  012737  023400  032100      MOV     #TST45,SKAD             ;IN CASE THE TEST ABORTS.
4478
4479  023242  113737  001102  001224      MOVB    $TSTNM,$TMP0
4480
4481  023250  104430                      SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4482  023252  104432                      SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4483  023254  104434                      SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4484  023256  104436                      SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
```

```
 4485  023260  012737  000014  177746          MOV     #M0M1,@#CONTRL           ;FORCE MISSES TO BOTH GROUPS.
 4486  023266  052737  000001  177746          BIS     #BIT0,@#CONTRL           ;DISABLE 'WARNING' TRAPS.
 4487  023274  012737  023336  000114          MOV     #KVERR,@#CACHVEC                 ;SET UP FOR THE ERROR ABOUT TO BE FORCED
 4488  023302  012704  040000                  MOV     #40000,R4                ;PATTERN FOR THE MAINTENANCE
 4489  023306  012702  177750                  MOV     #MAINT,R2                ;REGISTER.
 4490  023312  000402                          BR      KV1
 4491
 4492          023314                          LOC=.                    ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 4493          023314                          LOC=-4&LOC
 4494          023320                          LOC=LOC+4
 4495          023320                          .=LOC
 4496
 4497  023320  000240                  KV1:    NOP
 4498  023322  010412                          MOV     R4,(R2)                  ;SET THE MAINTENANCE REGISTER
 4499  023324  000240                          NOP                              ;WHEN THIS NOP IS FETCHED AN ERROR
 4500  023326  005701                  KV2:    TST     R1                       ;WILL BE RECOGNIZED BECAUSE OF THE
 4501                                                                           ;CONTENTS OF THE LOCATION KV2!
 4502                                                                           ;THIS PARITY ERROR WOULD
 4503                                                                           ;NORMALLY RELUT IN A TRAP BUT
 4504                                                                           ;BECAUSE TRAPS HAVE BEEN DISABLED
 4505                                                                           ;NONE SHOULD OCCUR!!!
 4506  023330  005012                          CLR     (R2)
 4507  023332  000240                          NOP
 4508  023334  000420                          BR      KVDONE                   ;GOOD, NO TRAP OCCURRED!
 4509
 4510  023336                          KVERR:                           ;COME HERE IF A TRAP OCCURS
 4511  023336  012637  001230                  MOV     (SP)+,$TMP2              ;AND REPORT THE ERROR.
 4512  023342  005726                          TST     (SP)+
 4513  023344  013737  177746  001232          MOV     @#CONTRL,$TMP3
 4514  023352  013737  177740  001234          MOV     @#LOADRS,$TMP4
 4515  023360  013737  177742  001236          MOV     @#HIADRS,$TMP5
 4516  023366  013737  177744  001240          MOV     @#MEMERR,$TMP6
 4517  023374  104134                  1$:     ERROR   134
 4518
 4519  023376  104416                  KVDONE: RSET
 4520
 4521
 4522
 4523                                          ;;*************************************************************************
 4524                                          ;*TEST 45       CACHE CONTROL REGISTER DISABLE TRAPS TEST 2
 4525                                          ;*
 4526                                          ;*THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION.
 4527                                          ;*IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE ADDRESS
 4528                                          ;*MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO
 4529                                          ;*FORCE THE ERROR ON THE LOW BYTE OF THE ADDRESS, IN THE ADDRESS MEMORY
 4530                                          ;*OF GROUP 0.
 4531                                          ;*
 4532                                          ;;*************************************************************************
 4533  023400  000004                  TST45:  SCOPE
 4534  023402  012737  000040  001274          MOV     #40,$TIMES       ;;DO 40 ITERATIONS
 4535          000045                  KX=$TN-1
 4536                                                                   ;SET THE SKAD REGISTER
 4537  023410  012737  023600  032100          MOV     #TST46,SKAD      ;IN CASE THE TEST ABORTS.
 4538
 4539  023416  113737  001102  001224          MOVB    $TSTNM,$TMP0
 4540
```

```
4541  023424  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4542  023426  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4543  023430  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4544  023432  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4545  023434  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;USE GROUP ZERO
4546  023442  012700  023530                  MOV     #KX2,R0         ;MAKE KX2 A HIT IN GROUP
4547  023446  005710                          TST     (R0)            ;ZERO.
4548  023450  005710                          TST     (R0)
4549
4550                                                                  ;SEE IF REFERENCE ADDRESS
4551  023452  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
4552  023460  001007                          BNE     KX1
4553                                                                  ;IF NOT ERROR!
4554  023462  010037  001230                  MOV     R0,$TMP2
4555  023466  012737  000000  001226          MOV     #0,$TMP1
4556  023474  104001                          ERROR   1
4557
4558  023476  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
4559
4560  023500  052737  000001  177746  KX1:    BIS     #BIT0,@#CONTRL  ;DISABLE 'WARNING' TRAPS.
4561  023506  012737  023536  000114          MOV     #KXERR,@#CACHVEC        ;SET UP FOR ERROR WHICH
4562                                                                  ;SHOULD NOT TRAP!
4563  023514  012704  000400                  MOV     #400,R4         ;PATTERN FOR MAINT REG.
4564  023520  012702  177750                  MOV     #MAINT,R2
4565  023524  000240                          NOP
4566  023526  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
4567  023530  005012                  KX2:    CLR     (R2)            ;THE FETCH OF THIS
4568  023532  000240                          NOP                     ;INSTRUCTION SHOULD CAUSE
4569  023534  000420                          BR      KXDONE          ;A CACHE  MEMORY
4570                                                                  ;PARITY ERROR WHICH
4571                                                                  ;NORMALLY SHOULD TRAP
4572                                                                  ;BUT HERE NO TRAP SHOULD
4573                                                                  ;OCCUR FOR TRAPS HAVE BEEN DISABLED.
4574
4575  023536                          KXERR:                          ;A TRAP HAS ERRONEOUSLY
4576  023536  012637  001230                  MOV     (SP)+,$TMP2     ;TAKEN PLACE, REPORT
4577  023542  005726                          TST     (SP)+           ;UNABLE TO DISABLE TRAPS.
4578  023544  013737  177746  001232          MOV     @#CONTRL,$TMP3
4579  023552  013737  177740  001234          MOV     @#LOADRS,$TMP4
4580  023560  013737  177742  001236          MOV     @#HIADRS,$TMP5
4581  023566  013737  177744  001240          MOV     @#MEMERR,$TMP6
4582
4583  023574  104134                  1$:     ERROR   134
4584
4585  023576  104416                  KXDONE: RSET
4586
4587
4588                                  ;;*****************************************************************
4589                                  ;*TEST 46        CACHE CONTROL REGISTER DISABLE TRAPS TEST 3
4590                                  ;*
4591                                  ;*THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION.
4592                                  ;*IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE
4593                                  ;*MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO
4594                                  ;*FORCE THE ERROR ON THE LOW BYTE OF THE . IN THE  MEMORY
4595                                  ;*OF GROUP 0.
4596                                  ;*
```

```
                                ;;*****************************************************************
023600  000004                  TST46:  SCOPE
023602  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000046                  KZ=$TN-1
                                                                ;SET THE SKAD REGISTER
023610  012737  024000  032100          MOV     #TST47,SKAD     ;IN CASE THE TEST ABORTS.

023616  113737  001102  001224          MOVB    $TSTNM,$TMP0

023624  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
023626  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
023630  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
023632  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
023634  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;USE GROUP ZERO
023642  012700  023730                  MOV     #KZ2,R0         ;MAKE KZ2 A HIT IN GROUP
023646  005710                          TST     (R0)            ;ZERO.
023650  005710                          TST     (R0)

                                                                ;SEE IF REFERENCE ADDRESS
023652  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
023660  001007                          BNE     KZ1
                                                                ;IF NOT ERROR:
023662  010037  001230                  MOV     R0,$TMP2
023666  012737  000000  001226          MOV     #0,$TMP1
023674  104001                          ERROR   1

023676  104420                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

023700  052737  000001  177746  KZ1:    BIS     #BIT0,@#CONTRL  ;DISABLE 'WARNING' TRAPS.
023706  012737  023736  000114          MOV     #KZERR,@#CACHVEC        ;SET UP FOR ERROR WHICH
                                                                ;SHOULD NOT TRAP!
023714  012704  000020                  MOV     #20,R4          ;PATTERN FOR MAINT REG.
023720  012702  177750                  MOV     #MAINT,R2
023724  000240                          NOP
023726  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
023730  005012                  KZ2:    CLR     (R2)            ;THE FETCH OF THIS
023732  000240                          NOP                     ;INSTRUCTION SHOULD CAUSE
023734  000420                          BR      KZDONE          ;A CACHE  MEMORY
                                                                ;PARITY ERROR WHICH
                                                                ;NORMALLY SHOULD TRAP
                                                                ;BUT HERE NO TRAP SHOULD
                                                                ;OCCUR FOR TRAPS HAVE BEEN DISABLED.

023736                          KZERR:                          ;A TRAP HAS ERRONEOUSLY
023736  012637  001230                  MOV     (SP)+,$TMP2     ;TAKEN PLACE, REPORT
023742  005726                          TST     (SP)+           ;UNABLE TO DISABLE TRAPS.
023744  013737  177746  001232          MOV     @#CONTRL,$TMP3
023752  013737  177740  001234          MOV     @#LOADRS,$TMP4
023760  013737  177742  001236          MOV     @#HIADRS,$TMP5
023766  013737  177744  001240          MOV     @#MEMERR,$TMP6

023774  104134                  1$:     ERROR   134

023776  104416                  KZDONE: RSET
```

```
                                        ;;*************************************************************************
                                        ;*TEST 47        CACHE ERROR REGISTER LOCK UP TEST 1
                                        ;*
                                        ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
                                        ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
                                        ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
                                        ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
                                        ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
                                        ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
                                        ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
                                        ;*TO THE CACHE DIRECTLY.
                                        ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
                                        ;*TO THE CACHE DIRECTLY.
                                        ;*
                                        ;;*************************************************************************
024000  000004                  TST47:  SCOPE
024002  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000047          NA=$TN-1
                                                                ;SET THE SKAD REGISTER
024010  012737  024364  032100          MOV     #TST50,SKAD     ;IN CASE THE TEST ABORTS.

024016  113737  001102  001224          MOVB    $TSTNM,$TMP0

024024  104430                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
024026  104432                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
024030  104434                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
024032  104436                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
024034  012737  000014  177746          MOV     #MOM1,@#CNTRL           ;FORCE MISSES TO BOTH GROUPS.


024042  012737  024116  000114          MOV     #NA3,@#CACHVEC          ;SET UP FOR THE ERROR.
024050  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
024054  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
024060  000401                          BR      NA1

        024062                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        024060                          LOC=-4&LOC
        024064                          LOC=LOC+4
        024064                          .=LOC

024064  000240                  NA1:    NOP
024066  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
024070  005701                  NA2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
024072  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
024074  000240                          NOP
                                                                ;IF NONE OCCURS REPORT
024076  012737  010000  001230          MOV     #10000,$TMP2    ;ERROR!
024104  104127                  1$:     ERROR   127
024106  012737  177777  032334          MOV     #-1,MANFL2
024114  000522                  BR      NADONE


024116                  NA3:
```

```
4709
4710  024116  012737  024172  000114          MOV     #NA6,@#CACHVEC          ;SET UP FOR THE ERROR.
4711  024124  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
4712  024130  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
4713  024134  000401                          BR      NA4
4714
4715          024136                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4716          024134                          LOC=-4&LOC
4717          024140                          LOC=LOC+4
4718          024140                          .=LOC
4719
4720  024140  000240                  NA4:    NOP
4721  024142  010412                          MOV     R4,(R2)                 ;SET THE MAINT. REG.
4722  024144  005701                  NA5:    TST     R1                      ;THE FETCH OF THIS INSTRUCTION
4723  024146  005012                          CLR     (R2)                    ;SHOULD CAUSE AN ABORT!
4724  024150  000240                          NOP
4725                                                                          ;IF NONE OCCURS REPORT
4726  024152  012737  010000  001230          MOV     #10000,$TMP2            ;ERROR!
4727  024160  104127                  1$:     ERROR   127
4728  024162  012737  177777  032334          MOV     #-1,MANFL2
4729  024170  000474                  BR      NADONE
4730
4731
4732  024172                          NA6:
4733
4734  024172  062706  000010                  ADD     #10,SP                  ;RESET THE STACK.
4735  024176  022737  144404  177744          CMP     #144404,@#MEMERR                ;SEE IF THE ERROR REGISTER
4736  024204  001004                          BNE     NA7                     ;IS SET CORRECTLY.
4737  024206  022737  024070  177740          CMP     #NA2,@#LOADRS           ;SEE IF THE ADDRESS REGISTER
4738  024214  001422                          BEQ     NA8                     ;IS SET CORRECTLY.
4739
4740  024216                          NA7:                                    ;NOT SET CORRECTLY!
4741  024216  012737  144404  001230          MOV     #144404,$TMP2           ;REPORT FAILURE.
4742  024224  013737  177744  001232          MOV     @#MEMERR,$TMP3
4743  024232  012737  024070  001234          MOV     #NA2,$TMP4
4744  024240  005037  001236                  CLR     $TMP5
4745  024244  013737  177740  001240          MOV     @#LOADRS,$TMP6
4746  024252  013737  177742  001242          MOV     @#HIADRS,$TMP7
4747
4748  024260  104135                  1$:     ERROR   135
4749
4750  024262  005037  177572          NA8:    CLR     @#MMR0                  ;TURN OFF MEMORY MANAGEMENT.
4751  024266  005037  172516                  CLR     @#MMR3
4752  024272  012737  177777  177744          MOV     #-1,@#MEMERR            ;SEE IF YOU CAN CLR THE
4753  024300  005737  177744                  TST     @#MEMERR                ;ERROR REG.
4754  024304  001416                          BEQ     NA10
4755
4756  024306                          NA9:                                    ;WON'T CLEAR!
4757  024306  013737  177740  001230          MOV     @#LOADRS,$TMP2
4758  024314  013737  177742  001232          MOV     @#HIADRS,$TMP3
4759  024322  013737  177744  001234          MOV     @#MEMERR,$TMP4
4760
4761  024330  104130                  1$:     ERROR   130
4762  024332  012737  177777  032314          MOV     #-1,MMRFLG
4763  024340  000410                          BR      NADONE
4764
```

```
4765  024342  022737  177740  177740  NA10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
4766  024350  001356                          BNE     NA9              ;HAS RESET
4767  024352  022737  000003  177742          CMP     #3,@#HIADRS
4768  024360  001352                          BNE     NA9
4769
4770  024362  104416                  NADONE: RSET
4771
4772
4773                                  ;;*********************************************************************
4774                                  ;*TEST 50        CACHE ERROR REGISTER LOCK UP TEST 2
4775                                  ;*
4776                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
4777                                  ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
4778                                  ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
4779                                  ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
4780                                  ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
4781                                  ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
4782                                  ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
4783                                  ;*TO THE CACHE DIRECTLY.
4784                                  ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
4785                                  ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
4786                                  ;*
4787                                  ;;*********************************************************************
4788  024364  000004                  TST50:  SCOPE
4789  024366  012737  000040  001274          MOV     #40,$TIMES       ;;DO 40 ITERATIONS
4790          000050                  NB=$TN-1
4791                                                                   ;SET THE SKAD REGISTER
4792  024374  012737  025054  032100          MOV     #TST51,SKAD      ;IN CASE THE TEST ABORTS.
4793
4794  024402  113737  001102  001224          MOVB    $TSTNM,$TMP0
4795
4796  024410  104430                          SKPBER                   ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4797  024412  104432                          SKPBCN                   ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4798  024414  104434                          SKPBMN                   ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4799  024416  104436                          SKPBHM                   ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4800  024420  104422                          MMSKIP
4801
4802  024422  012700  172340                  MOV     #KIPAR0,R0       ;SET UP MEMORY MANAGEMENT
4803                                                                   ;TO RELOCATE EVERYTHING
4804  024426  012702  172300                  MOV     #KIPDR0,R2       ;THROUGH THE UNIBUS
4805  024432  012703  000007                  MOV     #7,R3            ;MAP PASSIVELY TO MEMORY.
4806  024436  005004                          CLR     R4               ;BY PASSIVELY IS MEANT
4807  024440  012705  170200                  MOV     #MAPL00,R5       ;THAT ADDRESS ARE
4808                                                                   ;RELOCATED TO THEMSELVES.
4809  024444  012722  077406          64$:    MOV     #77406,(R2)+
4810  024450  010401                          MOV     R4,R1
4811  024452  072127  000006                  ASH     #6,R1
4812  024456  010125                          MOV     R1,(R5)+
4813  024460  005025                          CLR     (R5)+
4814  024462  010410                          MOV     R4,(R0)
4815  024464  062720  170000                  ADD     #170000,(R0)+
4816  024470  062704  000200                  ADD     #200,R4
4817  024474  077315                          SOB     R3,64$
4818  024476  012710  177600                  MOV     #177600,(R0)
4819  024502  012712  077406                  MOV     #77406,(R2)
4820
```

```
 4821  024506  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
 4822
 4823
 4824  024514  012737  024572  000114          MOV     #NB3,@#CACHVEC          ;SET UP FOR THE ERROR.
 4825  024522  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
 4826  024526  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
 4827  024532  000402                          BR      NB1
 4828
 4829          024534                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 4830          024534                          LOC=-4&LOC
 4831          024540                          LOC=LOC+4
 4832          024540                          .=LOC
 4833
 4834  024540  000240                  NB1:    NOP
 4835  024542  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
 4836  024544  005701                  NB2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
 4837  024546  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
 4838  024550  000240                          NOP
 4839                                                                  ;IF NONE OCCURS REPORT
 4840  024552  012737  010000  001230          MOV     #10000,$TMP2    ;ERROR!
 4841  024560  104127                  1$:     ERROR   127
 4842  024562  012737  177777  032334          MOV     #-1,MANFL2
 4843  024570  000530                  BR      NBDONE
 4844
 4845
 4846  024572                          NB3:
 4847
 4848  024572  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
 4849  024600  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
 4850  024606  012737  024662  000114          MOV     #NB6,@#CACHVEC  ;SET UP FOR ERROR
 4851  024614  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
 4852  024620  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
 4853  024624  000401                          BR      NB4
 4854
 4855          024626                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 4856          024624                          LOC=-4&LOC
 4857          024630                          LOC=LOC+4
 4858          024630                          .=LOC
 4859
 4860  024630  000240                  NB4:    NOP
 4861  024632  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
 4862  024634  005701                  NB5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
 4863  024636  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
 4864  024640  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
 4865                                                                  ;NO ABORT OCCURRED!
 4866  024642  012737  010000  001230          MOV     #10000,$TMP2    ;REPORT FAILURE
 4867  024650  104127                  1$:     ERROR   127
 4868  024652  012737  177777  032320          MOV     #-1,MANFLG
 4869  024660  000474                          BR      NBDONE
 4870
 4871
 4872  024662                          NB6:
 4873
 4874  024662  062706  000010                  ADD     #10,SP          ;RESET THE STACK.
 4875  024666  022737  137404  177744          CMP     #137404,@#MEMERR        ;SEE IF THE ERROR REGISTER
 4876  024674  001004                          BNE     NB7             ;IS SET CORRECTLY.
```

```
4877  024676  022737  024544  177740          CMP     #NB2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
4878  024704  001422                          BEQ     NB8             ;IS SET CORRECTLY.
4879
4880  024706                          NB7:                            ;NOT SET CORRECTLY!
4881  024706  012737  137404  001230          MOV     #137404,$TMP2   ;REPORT FAILURE.
4882  024714  013737  177744  001232          MOV     @#MEMERR,$TMP3
4883  024722  012737  024544  001234          MOV     #NB2,$TMP4
4884  024730  005037  001236                  CLR     $TMP5
4885  024734  013737  177740  001240          MOV     @#LOADRS,$TMP6
4886  024742  013737  177742  001242          MOV     @#HIADRS,$TMP7
4887
4888  024750  104135                  1$:     ERROR   135
4889
4890  024752  005037  177572          NB8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
4891  024756  005037  172516                  CLR     @#MMR3
4892  024762  012737  177777  177744          MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
4893  024770  005737  177744                  TST     @#MEMERR        ;ERROR REG.
4894  024774  001416                          BEQ     NB10
4895
4896  024776                          NB9:                            ;WON'T CLEAR!
4897  024776  013737  177740  001230          MOV     @#LOADRS,$TMP2
4898  025004  013737  177742  001232          MOV     @#HIADRS,$TMP3
4899  025012  013737  177744  001234          MOV     @#MEMERR,$TMP4
4900
4901  025020  104130                  1$:     ERROR   130
4902  025022  012737  177777  032314          MOV     #-1,MMRFLG
4903  025030  000410                          BR      NBDONE
4904
4905  025032  022737  177740  177740  NB10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
4906  025040  001356                          BNE     NB9             ;HAS RESET
4907  025042  022737  000003  177742          CMP     #3,@#HIADRS
4908  025050  001352                          BNE     NB9
4909
4910  025052  104416                  NBDONE: RSET
4911
4912
4913                                  ;;*******************************************************************
4914                                  ;*TEST 51       CACHE ERROR REGISTER LOCK UP TEST 3
4915                                  ;*
4916                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
4917                                  ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
4918                                  ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
4919                                  ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
4920                                  ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
4921                                  ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
4922                                  ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
4923                                  ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
4924                                  ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
4925                                  ;*TO THE CACHE DIRECTLY.
4926                                  ;*
4927                                  ;;*******************************************************************
4928  025054  000004                  TST51:  SCOPE
4929  025056  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4930          000051                  NC=$TN-1
4931                                                                  ;SET THE SKAD REGISTER
4932  025064  012737  025554  032100          MOV     #TST52,SKAD     ;IN CASE THE TEST ABORTS.
```

```
4933
4934  025072  113737  001102  001224          MOVB    $TSTNM,$TMP0
4935
4936  025100  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4937  025102  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4938  025104  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4939  025106  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4940  025110  104422                          MMSKIP
4941
4942  025112  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
4943                                                                  ;TO RELOCATE EVERYTHING
4944  025116  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
4945  025122  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
4946  025126  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
4947  025130  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
4948                                                                  ;RELOCATED TO THEMSELVES.
4949  025134  012722  077406          64$:    MOV     #77406,(R2)+
4950  025140  010401                          MOV     R4,R1
4951  025142  072127  000006                  ASH     #6,R1
4952  025146  010125                          MOV     R1,(R5)+
4953  025150  005025                          CLR     (R5)+
4954  025152  010410                          MOV     R4,(R0)
4955  025154  062720  170000                  ADD     #170000,(R0)+
4956  025160  062704  000200                  ADD     #200,R4
4957  025164  077315                          SOB     R3,64$
4958  025166  012710  177600                  MOV     #177600,(R0)
4959  025172  012712  077406                  MOV     #77406,(R2)
4960
4961  025176  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
4962
4963
4964  025204  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
4965  025212  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
4966  025220  012737  025276  000114          MOV     #NC3,@#CACHVEC  ;SET UP FOR ERROR
4967  025226  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
4968  025232  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
4969  025236  000402                          BR      NC1
4970
4971          025240                          LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4972          025240                          LOC=-4&LOC
4973          025244                          LOC=LOC+4
4974          025244                          .=LOC
4975
4976  025244  000240                  NC1:    NOP
4977  025246  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
4978  025250  005701                  NC2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
4979  025252  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
4980  025254  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
4981                                                                  ;NO ABORT OCCURRED!
4982  025256  012737  010000  001230          MOV     #10000,$TMP2    ;REPORT FAILURE
4983  025264  104127                  1$:     ERROR   127
4984  025266  012737  177777  032320          MOV     #-1,MANFLG
4985  025274  000526                          BR      NCDONE
4986
4987
4988  025276  005037  177572          NC3:    CLR     @#MMR0                  ;TURN OFF MEMORY MANAGEMENT.
```

```
 4989  025302  005037  172516                  CLR     @#MMR3
 4990
 4991  025306  012737  025362  000114          MOV     #NC6,@#CACHVEC          ;SET UP FOR THE ERROR.
 4992  025314  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
 4993  025320  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
 4994  025324  000401                          BR      NC4
 4995
 4996          025326                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 4997          025324                          LOC=-4&LOC
 4998          025330                          LOC=LOC+4
 4999          025330                          .=LOC
 5000
 5001  025330  000240                  NC4:    NOP
 5002  025332  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
 5003  025334  005701                  NC5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
 5004  025336  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
 5005  025340  000240                          NOP
 5006                                                                  ;IF NONE OCCURS REPORT
 5007  025342  012737  010000  001230          MOV     #10000,$TMP2    ;ERROR!
 5008  025350  104127                  1$:     ERROR   127
 5009  025352  012737  177777  032334          MOV     #-1,MANFL2
 5010  025360  000474                  BR      NCDONE
 5011
 5012
 5013  025362                          NC6:
 5014
 5015  025362  062706  000010                  ADD     #10,SP          ;RESET THE STACK.
 5016  025366  022737  167404  177744          CMP     #167404,@#MEMERR        ;SEE IF THE ERROR REGISTER
 5017  025374  001004                          BNE     NC7             ;IS SET CORRECTLY.
 5018  025376  022737  025250  177740          CMP     #NC2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
 5019  025404  001422                          BEQ     NC8             ;IS SET CORRECTLY.
 5020
 5021  025406                          NC7:                            ;NOT SET CORRECTLY!
 5022  025406  012737  167404  001230          MOV     #167404,$TMP2   ;REPORT FAILURE.
 5023  025414  013737  177744  001232          MOV     @#MEMERR,$TMP3
 5024  025422  012737  025250  001234          MOV     #NC2,$TMP4
 5025  025430  005037  001236                  CLR     $TMP5
 5026  025434  013737  177740  001240          MOV     @#LOADRS,$TMP6
 5027  025442  013737  177742  001242          MOV     @#HIADRS,$TMP7
 5028
 5029  025450  104135                  1$:     ERROR   135
 5030
 5031  025452  005037  177572          NC8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
 5032  025456  005037  172516                  CLR     @#MMR3
 5033  025462  012737  177777  177744          MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
 5034  025470  005737  177744                  TST     @#MEMERR        ;ERROR REG.
 5035  025474  001416                          BEQ     NC10
 5036
 5037  025476                          NC9:                            ;WON'T CLEAR!
 5038  025476  013737  177740  001230          MOV     @#LOADRS,$TMP2
 5039  025504  013737  177742  001232          MOV     @#HIADRS,$TMP3
 5040  025512  013737  177744  001234          MOV     @#MEMERR,$TMP4
 5041
 5042  025520  104130                  1$:     ERROR   130
 5043  025522  012737  177777  032314          MOV     #-1,MMRFLG
 5044  025530  000410                          BR      NCDONE
```

```
5045
5046  025532  022737  177740  177740  NC10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
5047  025540  001356                          BNE     NC9              ;HAS RESET
5048  025542  022737  000003  177742          CMP     #3,@#HIADRS
5049  025550  001352                          BNE     NC9
5050
5051  025552  104416                  NCDONE: RSET
5052
5053
5054                                  ;;*******************************************************************
5055                                  ;*TEST 52       CACHE ERROR REGISTER LOCK UP TEST 4
5056                                  ;*
5057                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
5058                                  ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
5059                                  ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
5060                                  ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
5061                                  ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
5062                                  ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
5063                                  ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
5064                                  ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
5065                                  ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
5066                                  ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
5067                                  ;*
5068                                  ;*******************************************************************
5069  025554  000004                  TST52:  SCOPE
5070  025556  012737  000040  001274          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
5071           000052                 ND=$TN-1
5072                                                  ;SET THE SKAD REGISTER
5073  025564  012737  026260  032100          MOV     #TST53,SKAD     ;IN CASE THE TEST ABORTS.
5074
5075  025572  113737  001102  001224          MOVB    $TSTNM,$TMP0
5076
5077  025600  104430                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
5078  025602  104432                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
5079  025604  104434                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
5080  025606  104436                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
5081  025610  104422                          MMSKIP
5082
5083  025612  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
5084                                                                  ;TO RELOCATE EVERYTHING
5085  025616  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
5086  025622  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
5087  025626  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
5088  025630  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
5089                                                                  ;RELOCATED TO THEMSELVES.
5090  025634  012722  077406          64$:    MOV     #77406,(R2)+
5091  025640  010401                          MOV     R4,R1
5092  025642  072127  000006                  ASH     #6,R1
5093  025646  010125                          MOV     R1,(R5)+
5094  025650  005025                          CLR     (R5)+
5095  025652  010410                          MOV     R4,(R0)
5096  025654  062720  170000                  ADD     #170000,(R0)+
5097  025660  062704  000200                  ADD     #200,R4
5098  025664  077315                          SOB     R3,64$
5099  025666  012710  177600                  MOV     #177600,(R0)
5100  025672  012712  077406                  MOV     #77406,(R2)
```

```
5101
5102  025676  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
5103
5104
5105  025704  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
5106  025712  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
5107  025720  012737  025776  000114          MOV     #ND3,@#CACHVEC  ;SET UP FOR ERROR
5108  025726  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
5109  025732  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
5110  025736  000402                          BR      ND1
5111
5112          025740                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5113          025740                          LOC=-4&LOC
5114          025744                          LOC=LOC+4
5115          025744                          .=LOC
5116
5117  025744  000240                  ND1:    NOP
5118  025746  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
5119  025750  005701                  ND2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
5120  025752  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
5121  025754  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
5122                                                                  ;NO ABORT OCCURRED!
5123  025756  012737  010000  001230          MOV     #10000,$TMP2    ;REPORT FAILURE
5124  025764  104127                  1$:     ERROR   127
5125  025766  012737  177777  032320          MOV     #-1,MANFLG
5126  025774  000530                          BR      NDDONE
5127
5128
5129  025776                          ND3:
5130
5131  025776  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
5132  026004  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
5133  026012  012737  026066  000114          MOV     #ND6,@#CACHVEC  ;SET UP FOR ERROR
5134  026020  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
5135  026024  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
5136  026030  000401                          BR      ND4
5137
5138          026032                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5139          026030                          LOC=-4&LOC
5140          026034                          LOC=LOC+4
5141          026034                          .=LOC
5142
5143  026034  000240                  ND4:    NOP
5144  026036  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
5145  026040  005701                  ND5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
5146  026042  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
5147  026044  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
5148                                                                  ;NO ABORT OCCURRED!
5149  026046  012737  010000  001230          MOV     #10000,$TMP2    ;REPORT FAILURE
5150  026054  104127                  1$:     ERROR   127
5151  026056  012737  177777  032320          MOV     #-1,MANFLG
5152  026064  000474                          BR      NDDONE
5153
5154
5155  026066                          ND6:
5156
```

```
5157  026066  062706  000010            ADD     #10,SP                  ;RESET THE STACK.
5158  026072  022737  033404  177744    CMP     #33404,@#MEMERR         ;SEE IF THE ERROR REGISTER
5159  026100  001004                    BNE     ND7                     ;IS SET CORRECTLY.
5160  026102  022737  025750  177740    CMP     #ND2,@#LOADRS           ;SEE IF THE ADDRESS REGISTER
5161  026110  001422                    BEQ     ND8                     ;IS SET CORRECTLY.
5162
5163  026112                    ND7:                                    ;NOT SET CORRECTLY!
5164  026112  012737  033404  001230    MOV     #33404,$TMP2            ;REPORT FAILURE.
5165  026120  013737  177744  001232    MOV     @#MEMERR,$TMP3
5166  026126  012737  025750  001234    MOV     #ND2,$TMP4
5167  026134  005037  001236            CLR     $TMP5
5168  026140  013737  177740  001240    MOV     @#LOADRS,$TMP6
5169  026146  013737  177742  001242    MOV     @#HIADRS,$TMP7
5170
5171  026154  104135              1$:   ERROR   135
5172
5173  026156  005037  177572    ND8:    CLR     @#MMR0                  ;TURN OFF MEMORY MANAGEMENT.
5174  026162  005037  172516            CLR     @#MMR3
5175  026166  012737  177777  177744    MOV     #-1,@#MEMERR            ;SEE IF YOU CAN CLR THE
5176  026174  005737  177744            TST     @#MEMERR                ;ERROR REG.
5177  026200  001416                    BEQ     ND10
5178
5179  026202                    ND9:                                    ;WON'T CLEAR!
5180  026202  013737  177740  001230    MOV     @#LOADRS,$TMP2
5181  026210  013737  177742  001232    MOV     @#HIADRS,$TMP3
5182  026216  013737  177744  001234    MOV     @#MEMERR,$TMP4
5183
5184  026224  104130              1$:   ERROR   130
5185  026226  012737  177777  032314    MOV     #-1,MMRFLG
5186  026234  000410                    BR      NDDONE
5187
5188  026236  022737  177740  177740  ND10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
5189  026244  001356                    BNE     ND9             ;HAS RESET
5190  026246  022737  000003  177742          CMP     #3,@#HIADRS
5191  026254  001352                    BNE     ND9
5192
5193  026256  104416            NDDONE: RSET
5194
5195
5196                                    ;;*********************************************************************
5197                                    ;*TEST 53        MAIN MEMORY DATA PARITY CHECKERS LOW BYTE TEST
5198                                    ;*
5199                                    ;*THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS
5200                                    ;*FOR THE LOW BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD.
5201                                    ;*THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY
5202                                    ;*ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY
5203                                    ;*BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE
5204                                    ;*THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT
5205                                    ;*A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE
5206                                    ;*AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY
5207                                    ;*BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS
5208                                    ;*SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1).
5209                                    ;*THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA
5210                                    ;*PARITY CHECKERS WORKS IN SUCH A WAY AS TO
5211                                    ;*EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO
5212                                    ;*THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO
```

```
5213                                    ;*AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS
5214                                    ;*ALREADY ONE THEN NO ERROR OCCURS!
5215                                    ;*
5216                                    ;;*****************************************************************************
5217  026260  000004                    TST53:  SCOPE
5218  026262  012737  000020  001274            MOV     #20,$TIMES      ;;DO 20 ITERATIONS
5219          000054                    UA=$TN
5220                                                                    ;SET THE SKAD REGISTER
5221  026270  012737  026634  032100            MOV     #TST54,SKAD     ;IN CASE THE TEST ABORTS.
5222
5223  026276  113737  001102  001224            MOVB    $TSTNM,$TMP0
5224  026304  012737  031754  000114            MOV     #SPUR,@#CACHVEC
5225
5226  026312  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
5227  026320  005000                            CLR     R0              ;INITIALIZE
5228
5229  026322  012737  026322  001110  UA1:      MOV     #UA1,$LPERR
5230  026330  004737  032340                    JSR     PC,PARCNT       ;SEE IF THE CURRENT TEST
5231  026334  032702  000001                    BIT     #BIT0,R2        ;PATTERN HAS THE PARITY BIT
5232  026340  001002                            BNE     UA2             ;OFF, IF NOT GO TO NEXT
5233  026342  000137  026614                    JMP     UA7             ;PATTERN
5234
5235  026346  012737  026520  000114  UA2:      MOV     #UAER1,@#CACHVEC        ;SET UP FOR THE ERROR, EVEN WORD.
5236  026354  012704  010000                    MOV     #10000,R4       ;THIS IS A PATTERN WHICH
5237  026360  012702  177750                    MOV     #MAINT,R2       ;WHEN LOADED INTO THE
5238                                                                    ;MAINTENANCE REGISTER
5239                                                                    ;WILL FORCE AN ERROR ON
5240                                                                    ;THE MAIN MEMORY EVEN
5241  026364  012701  026514                    MOV     #UATMP1,R1      ;WORD LOW BYTE
5242  026370  010011                            MOV     R0,(R1)
5243  026372  010412                            MOV     R4,(R2)         ;SET THE MAINT REG
5244  026374  021101                            CMP     (R1),R1         ;THE REFERENCE TO (R1),
5245                                                                    ;UATMP1 SHOULD CAUSE
5246                                                                    ;AN ERROR.
5247  026376  005012                            CLR     (R2)
5248  026400  005012                            CLR     (R2)
5249
5250  026402                            UA3:
5251                                                                    ;THE ERROR DIDN'T OCCUR!
5252  026402  010037  001230                    MOV     R0,$TMP2        ;REPORT FAILURE
5253  026406  012737  026514  001232            MOV     #UATMP1,$TMP3
5254  026414  005037  001234                    CLR     $TMP4
5255  026420  104140                    64$:    ERROR   140
5256
5257  026422  012737  026560  000114  UA4:      MOV     #UAER2,@#CACHVEC        ;SET UP FOR THE ERROR
5258  026430  012737  026422  001110            MOV     #UA4,$LPERR     ;ON THE ODD WORD.
5259  026436  012704  040000                    MOV     #40000,R4       ;THIS IS A PATTERN WHICH
5260  026442  012702  177750                    MOV     #MAINT,R2       ;WHEN LOADED IN THE MAINTENANCE
5261                                                                    ;REGISTER WILL CAUSE AN ERROR
5262  026446  012701  026516                    MOV     #UATMP2,R1      ;ON THE ODD WORD, LOW BYTE.
5263  026452  010011                            MOV     R0,(R1)         ;SET THE MAINT REG. AND
5264  026454  000240                            NOP
5265  026456  010412                            MOV     R4,(R2)         ;REFERENCE (R1), UATMP2, AND
5266  026460  021101                            CMP     (R1),R1         ;CAUSE THE ERROR.
5267
5268  026462  005012                            CLR     (R2)
```

```
5269  026464  005012                          CLR     (R2)
5270
5271  026466                          UA5:
5272                                                            ;THE ERROR DIDN'T OCCUR!
5273  026466  010037  001230                  MOV     R0,$TMP2        ;REPORT FAILURE
5274  026472  012737  026516  001232          MOV     #UATMP2,$TMP3
5275  026500  005037  001234                  CLR     $TMP4
5276  026504  104141                  64$:    ERROR   141
5277
5278  026506  000442                  UA6:    BR      UA7
5279
5280
5281          026510                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5282          026510                          LOC=-4&LOC
5283          026514                          LOC=LOC+4
5284          026514                          .=LOC
5285
5286  026514  000000                  UATMP1:.WORD    0
5287  026516  000000                  UATMP2:.WORD    0
5288
5289  026520                          UAER1:
5290  026520  022737  104404  177744          CMP     #104404,@#MEMERR        ;MAKE SURE THE ERROR
5291  026526  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
5292  026530  000137  031754          1$:     JMP     SPUR
5293  026534  022737  026514  177740  2$:     CMP     #UATMP1,@#LOADRS        ;MAKE SURE THE ERROR
5294  026542  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
5295                                                                  ;ADDRESS.
5296  026544  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
5297  026546  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
5298  026554  000137  026422                  JMP     UA4             ;GO TEST THE ODD WORD
5299
5300  026560                          UAER2:
5301  026560  022737  104410  177744          CMP     #104410,@#MEMERR        ;MAKE SURE THE ERROR
5302  026566  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
5303  026570  000137  031754          1$:     JMP     SPUR
5304  026574  022737  026516  177740  2$:     CMP     #UATMP2,@#LOADRS        ;MAKE SURE THE ERROR
5305  026602  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
5306                                                                  ;ADDRESS.
5307  026604  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
5308  026606  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
5309
5310  026614  022700  000377          UA7:    CMP     #377,R0         ;INCREMENT THE TEST PATTERN
5311  026620  001404                          BEQ     UA8
5312  026622  062700  000001                  ADD     #1,R0
5313  026626  000137  026322                  JMP     UA1
5314
5315  026632  104416                  UA8:    RSET
5316
5317                                  ;;*******************************************************************************
5318                                  ;*TEST 54        MAIN MEMORY DATA PARITY CHECKERS HIGH BYTE TEST
5319                                  ;*
5320                                  ;*THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS
5321                                  ;*FOR THE HIGH BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD.
5322                                  ;*THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY
5323                                  ;*ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY
5324                                  ;*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!
                                        ;*
                                        ;;*****************************************************************
026634  000004                  TST54:  SCOPE
026636  012737  000020  001274          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000055                  UB=$TN
                                                                ;SET THE SKAD REGISTER
026644  012737  027210  032100          MOV     #TST55,SKAD     ;IN CASE THE TEST ABORTS.

026652  113737  001102  001224          MOVB    $TSTNM,$TMP0
026660  012737  031754  000114          MOV     #SPUR,@#CACHVEC

026666  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
026674  005000                          CLR     R0              ;INITIALIZE

026676  012737  026676  001110  UB1:    MOV     #UB1,$LPERR
026704  004737  032340                  JSR     PC,PARCNT       ;SEE IF THE CURRENT TEST
026710  032702  000001                  BIT     #BIT0,R2        ;PATTERN HAS THE PARITY BIT
026714  001002                          BNE     UB2             ;OFF, IF NOT GO TO NEXT
026716  000137  027170                  JMP     UB7             ;PATTERN

026722  012737  027074  000114  UB2:    MOV     #UBER1,@#CACHVEC        ;SET UP FOR THE ERROR, EVEN WORD.
026730  012704  020000                  MOV     #20000,R4       ;THIS IS A PATTERN WHICH
026734  012702  177750                  MOV     #MAINT,R2       ;WHEN LOADED INTO THE
                                                                ;MAINTENANCE REGISTER
                                                                ;WILL FORCE AN ERROR ON
                                                                ;THE MAIN MEMORY EVEN
026740  012701  027070                  MOV     #UBTMP1,R1      ;WORD HIGH BYTE
026744  010011                          MOV     R0,(R1)
026746  010412                          MOV     R4,(R2)         ;SET THE MAINT REG
026750  021101                          CMP     (R1),R1         ;THE REFERENCE TO (R1),
                                                                ;UBTMP1 SHOULD CAUSE
                                                                ;AN ERROR.
026752  005012                          CLR     (R2)
026754  005012                          CLR     (R2)

026756                          UB3:
                                                                ;THE ERROR DIDN'T OCCUR!
026756  010037  001230                  MOV     R0,$TMP2        ;REPORT FAILURE
026762  012737  027070  001232          MOV     #UBTMP1,$TMP3
026770  005037  001234                  CLR     $TMP4
026774  104142                  64$:    ERROR   142

026776  012737  027134  000114  UB4:    MOV     #UBER2,@#CACHVEC        ;SET UP FOR THE ERROR
027004  012737  026776  001110          MOV     #UB4,$LPERR     ;ON THE ODD WORD.
027012  012704  100000                  MOV     #100000,R4      ;THIS IS A PATTERN WHICH
```

```
5381  027016  012702  177750              MOV     #MAINT,R2       ;WHEN LOADED IN THE MAINTENANCE
5382                                                              ;REGISTER WILL CAUSE AN ERROR
5383  027022  012701  027072              MOV     #UBTMP2,R1      ;ON THE ODD WORD, LOW BYTE.
5384  027026  010011                      MOV     R0,(R1)         ;SET THE MAINT REG. AND
5385  027030  000240                      NOP
5386  027032  010412                      MOV     R4,(R2)         ;REFERENCE (R1), UBTMP2, AND
5387  027034  021101                      CMP     (R1),R1         ;CAUSE THE ERROR.
5388
5389  027036  005012                      CLR     (R2)
5390  027040  005012                      CLR     (R2)
5391
5392  027042                      UB5:
5393                                                              ;THE ERROR DIDN'T OCCUR!
5394  027042  010037  001230              MOV     R0,$TMP2        ;REPORT FAILURE
5395  027046  012737  027072  001232      MOV     #UBTMP2,$TMP3
5396  027054  005037  001234              CLR     $TMP4
5397  027060  104143              64$:    ERROR   143
5398
5399  027062  000442              UB6:    BR      UB7
5400
5401
5402          027064                      LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5403          027064                      LOC=-4&LOC
5404          027070                      LOC=LOC+4
5405          027070                      .=LOC
5406
5407  027070  000000              UBTMP1:.WORD    0
5408  027072  000000              UBTMP2:.WORD    0
5409
5410  027074                      UBER1:
5411  027074  022737  104404  177744          CMP     #104404,@#MEMERR        ;MAKE SURE THE ERROR
5412  027102  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
5413  027104  000137  031754          1$:     JMP     SPUR
5414  027110  022737  027070  177740  2$:     CMP     #UBTMP1,@#LOADRS        ;MAKE SURE THE ERROR
5415  027116  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
5416                                                                  ;ADDRESS.
5417  027120  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
5418  027122  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
5419  027130  000137  026776                  JMP     UB4             ;GO TEST THE ODD WORD
5420
5421  027134                      UBER2:
5422  027134  022737  104410  177744          CMP     #104410,@#MEMERR        ;MAKE SURE THE ERROR
5423  027142  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY.
5424  027144  000137  031754          1$:     JMP     SPUR
5425  027150  022737  027072  177740  2$:     CMP     #UBTMP2,@#LOADRS        ;MAKE SURE THE ERROR
5426  027156  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
5427                                                                  ;ADDRESS.
5428  027160  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
5429  027162  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
5430
5431  027170  022700  177400      UB7:    CMP     #177400,R0              ;INCREMENT THE TEST PATTERN
5432  027174  001404                      BEQ     UB8
5433  027176  062700  000400              ADD     #400,R0
5434  027202  000137  026676              JMP     UB1
5435
5436  027206  104416              UB8:    RSET
```

```
                                TST55:
                                ;;*****************************************************************************
                                .SBTTL  END OF PASS ROUTINE
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                ;*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO LOOP
027210                          $EOP:
027210  000004                          SCOPE
027212  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
027216  005037  001274                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
027222  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
027226  042737  100000  001100        $EOPCT: BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
027234  005327                          DEC     (PC)+           ;;LOOP?
027236  000001                  $ENDCT: .WORD   1
027240  003031                          BGT     $DOAGN          ;;YES
027242  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
027244  000001                          .WORD   1
027246  027236                          $EOPCT
027250  104400  027330                  TYPE    ,$ENDMG         ;;TYPE 'END PASS #'
027254  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
027260  104410                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
027262  104400  027345                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
027266  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
027272  001414                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
027274  012703  125252                  MOV     #125252,R3
027300  004737  032414                  JSR     PC,CHAINQ
027304  013700  000042                  MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
027310  001405                          BEQ     $DOAGN          ;;RETURN ADDRESS
027312  000005                          RESET                   ;;CLEAR THE WORLD
027314  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
027316  000240                          NOP                     ;;SAVE ROOM
027320  000240                          NOP                     ;;FOR
027322  000240                          NOP                     ;;ACT11
027324                          $DOAGN:
027324  000137  004146                  JMP     @#LOOP          ;;RETURN
027330  005015  047105  020104  $ENDMG: .ASCIZ  <15><12>/END PASS #/
027336  040520  051523  021440
027344     000
027345     377      377      000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                ;;*****************************************************************************
                                .SBTTL  SCOPE HANDLER ROUTINE
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
```

```
5493                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
5494                                    ;*SW14=1        LOOP ON TEST
5495                                    ;*SW11=1        INHIBIT ITERATIONS
5496                                    ;*SW09=1        LOOP ON ERROR
5497                                    ;*SW08=1        LOOP ON TEST IN SWR<6:0>
5498                                    ;*CALL
5499                                    ;*      SCOPE           ;;SCOPE=IOT
5500
5501  027350                            $SCOPE:
5502  027350  006137  177570                   ROL     @#SWR           ;;LOOP ON PRESENT TEST?
5503  027354  100517                           BMI     $OVER           ;;YES IF SW14=1
5504                                    ;#####START OF CODE FOR THE XOR TESTER#####
5505  027356  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
5506                                                                    ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
5507  027360  013746  000004                   MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
5508  027364  012737  027404  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
5509  027372  005737  177060                   TST     @#177060        ;;TIME OUT ON XOR?
5510  027376  012637  000004                   MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
5511  027402  000471                           BR      $SVLAD          ;;GO TO THE NEXT TEST
5512  027404  022626                    5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
5513  027406  012637  000004                   MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
5514  027412  000431                           BR      7$              ;;LOOP ON THE PRESENT TEST
5515  027414                            6$:;#####END OF CODE FOR THE XOR TESTER#####
5516  027414  032737  000400  177570           BIT     #BIT08,@#SWR    ;;LOOP ON SPEC. TEST?
5517  027422  001412                           BEQ     2$              ;;BR IF NO
5518  027424  052737  001000  177746           BIS     #BIT9,  @#CONTRL        ;TURN OFF CACHE
5519  027432  013746  177570                   MOV     @#SWR,-(SP)     ;;SET DESIRED TEST NUM. FROM SWR
5520  027436  042716  000200                   BIC     #$SWRMK,(SP)    ;;STRIP AWAY UNDESIRED BITS
5521  027442  122637  001102                   CMPB    (SP)+,$TSTNM    ;;ON THE RIGHT TEST?
5522  027446  001462                           BEQ     $OVER           ;;BR IF YES
5523  027450  105737  001103            2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
5524  027454  001421                           BEQ     3$              ;;BR IF NO
5525  027456  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
5526  027464  101015                           BHI     3$              ;;BR IF NO
5527  027466  032737  001000  177570           BIT     #BIT09,@#SWR    ;;LOOP ON ERROR?
5528  027474  001404                           BEQ     4$              ;;BR IF NO
5529  027476  013737  001110  001106    7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
5530  027504  000443                           BR      $OVER
5531  027506  105037  001103            4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
5532  027512  005037  001274                   CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
5533  027516  000415                           BR      1$              ;;ESCAPE TO THE NEXT TEST
5534  027520  032737  004000  177570    3$:    BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
5535  027526  001011                           BNE     1$              ;;BR IF YES
5536  027530  005737  001100                   TST     $PASS           ;;IF FIRST PASS OF PROGRAM
5537  027534  001406                           BEQ     1$              ;;      INHIBIT ITERATIONS
5538  027536  005237  001104                   INC     $ICNT           ;;INCREMENT ITERATION COUNT
5539  027542  023737  001274  001104           CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
5540  027550  002021                           BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
5541  027552  012737  000001  001104    1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
5542  027560  013737  027630  001274           MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
5543  027566  105237  001102            $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
5544  027572  011637  001106                   MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
5545  027576  011637  001110                   MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
5546  027602  005037  001276                   CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
5547  027606  112737  000001  001115           MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
5548  027614  013737  001102  177570    $OVER: MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER
```

```
 5549  027622  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 5550  027626  000002                          RTI                     ;;FIXES PS
 5551  027630  000001                $MXCNT: 1                         ;;MAX. NUMBER OF ITERATIONS
 5552
 5553                                ;;**************************************************************
 5554
 5555                                .SBTTL  ERROR HANDLER ROUTINE
 5556
 5557                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 5558                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 5559                                ;*AND GO TO ERTYPE ON ERROR
 5560                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 5561                                ;*SW15=1        HALT ON ERROR
 5562                                ;*              HALT CAN OCCUR BEFORE AND AFTER THE ERROR TYPEOUT
 5563                                ;*SW13=1        INHIBIT ERROR TYPEOUTS
 5564                                ;*SW10=1        BELL ON ERROR
 5565                                ;*SW09=1        LOOP ON ERROR
 5566                                ;*CALL
 5567                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 5568
 5569  027632                        $ERROR:
 5570  027632  105237  001103        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 5571  027636  001775                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 5572  027640  013737  001102  177570        MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 5573  027646  005737  177570                TST     @#SWR           ;;HALT ON ERROR = 1?
 5574  027652  100001                        BPL     8$              ;;BRANCH IF NO
 5575  027654  000000                        HALT                    ;;YES--HALT
 5576  027656  032737  002000  177570  8$:   BIT     #BIT10,@#SWR    ;;BELL ON ERROR?
 5577  027664  001402                        BEQ     1$              ;;NO - SKIP
 5578  027666  104400  001300                TYPE    ,$BELL          ;;RING BELL
 5579  027672  005237  001112          1$:   INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 5580  027676  011637  001116                MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 5581  027702  162737  000002  001116        SUB     #2,$ERRPC
 5582  027710  117737  151202  001114        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 5583  027716  032737  020000  177570        BIT     #BIT13,@#SWR    ;;SKIP TYPEOUT IF SET
 5584  027724  001004                        BNE     2$              ;;SKIP TYPEOUTS
 5585  027726  004737  032616                JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
 5586  027732  104400  001305                TYPE    ,$CRLF
 5587  027736  005737  177570          2$:   TST     @#SWR           ;;HALT ON ERROR
 5588  027742  100001                        BPL     9$              ;;SKIP IF CONTINUE
 5589  027744  000000                        HALT                    ;;HALT ON ERROR!
 5590  027746  022737  027314  000042  9$:   CMP     #$ENDAD,42      ;;ACT-11?
 5591  027754  001001                        BNE     3$              ;;BRANCH IF NO
 5592  027756  000000                        HALT                    ;;YES
 5593  027760  032737  001000  177570  3$:   BIT     #BIT09,@#SWR    ;;LOOP ON ERROR SWITCH SET?
 5594  027766  001402                        BEQ     4$              ;;BR IF NO
 5595  027770  013716  001110                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 5596  027774  005737  001276          4$:   TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 5597  030000  001402                        BEQ     5$              ;;BR IF NONE
 5598  030002  013716  001276                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 5599  030006                          5$:
 5600  030006  012737  177777  177744        MOV     #-1,@#MEMERR
 5601  030014  005037  177766                CLR     @#CPUERR
 5602  030020  000002                        RTI
 5603
 5604                                ;;**************************************************************
```

```
 5605
 5606                                   .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
 5607
 5608                                   ;*SAVE R0-R5
 5609                                   ;*CALL:
 5610                                   ;*      SAVREG
 5611                                   ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
 5612                                   ;*
 5613                                   ;*TOP---(+16)
 5614                                   ;* +2---(+18)
 5615                                   ;* +4---R5
 5616                                   ;* +6---R4
 5617                                   ;* +8---R3
 5618                                   ;*+10---R2
 5619                                   ;*+12---R1
 5620                                   ;*+14---R0
 5621
 5622  030022                           $SAVREG:
 5623  030022  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 5624  030024  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5625  030026  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 5626  030030  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 5627  030032  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 5628  030034  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 5629  030036  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 5630  030042  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 5631  030046  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 5632  030052  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 5633  030056  000002                          RTI
 5634
 5635                                   ;*RESTORE R0-R5
 5636                                   ;*CALL:
 5637                                   ;*      RESREG
 5638  030060                           $RESREG:
 5639  030060  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 5640  030064  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 5641  030070  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 5642  030074  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 5643  030100  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 5644  030102  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 5645  030104  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 5646  030106  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 5647  030110  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 5648  030112  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 5649  030114  000002                          RTI
 5650
 5651                                   ;;*************************************************************************
 5652
 5653                                   .SBTTL  TYPE ROUTINE
 5654
 5655                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 5656                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 5657                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 5658                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 5659                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 5660                                   ;*
```

```
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*
                                ;*2) USING A JSR INSTRUCTION
                                ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
                                ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
                                ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE

030116  105737  001151          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
030122  100002                          BPL     1$              ;;BR IF YES
030124  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
030126  000407                          BR      3$              ;;LEAVE
030130  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
030132  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
030136  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
030140  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
030142  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
030144  012600                          MOV     (SP)+,R0        ;;RESTORE R0
030146  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
030152  000002                          RTI                     ;;RETURN
030154  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
030160  001426                          BEQ     8$
030162  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT
030166  001004                          BNE     5$
030170  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
030172  104400  001305                  TYPE    ,$CRLF
030176  000757                          BR      2$              ;;GET NEXT CHARACTER
030200  004737  030262          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
030204  123726  001150          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
030210  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
030212  013746  001146                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
030216  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
030222  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
030224  004737  030262                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
030230  105337  030326                  DECB    $CHARCNT        ;;DON'T COUNT THE NULL AS A CHARACTER
030234  000770                          BR      7$              ;;LOOP

                                ;;HORIZONTAL TAB PROCESSOR

030236  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
030242  004737  030262          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
030246  132737  000007  030326          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
030254  001372                          BNE     9$              ;;TAB STOP
030256  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
030260  000726                          BR      2$              ;;GET NEXT CHARACTER
030262  105777  150654          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
030266  100375                          BPL     $TYPEC
030270  116677  000002  150646          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
030276  122766  000015  000002          CMPB    #CR,2(SP)       ;;BRANCH IF
030304  001003                          BNE     1$              ;;NOT <CR>
030306  105037  030326                  CLRB    $CHARCNT        ;;
```

```
 5717  030312  000406                          BR      $TYPEX          ;;EXIT
 5718  030314  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;BRANCH IF
 5719  030322  001402                          BEQ     $TYPEX          ;;<LF>
 5720  030324  105227                          INCB    (PC)+           ;;INC SPACE
 5721  030326  000000                  $CHARCNT:.WORD  0               ;;COUNT
 5722  030330  000207                  $TYPEX: RTS     PC
 5723
 5724
 5725                                  ;;**************************************************************************
 5726
 5727                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 5728
 5729                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 5730                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 5731                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 5732                                  ;*CALL:
 5733                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 5734                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
 5735                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 5736                                  ;*      .BYTE   M               ;;M=1 OR 0
 5737                                  ;*                                      ;;1=TYPE LEADING ZEROS
 5738                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
 5739                                  ;*
 5740                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 5741                                  ;*$TYPOS OR $TYPOC
 5742                                  ;*CALL:
 5743                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 5744                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
 5745                                  ;*
 5746                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 5747                                  ;*CALL:
 5748                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 5749                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
 5750
 5751  030332  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 5752  030336  116637  000001  030555          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 5753  030344  112637  030557                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 5754  030350  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 5755  030354  000406                          BR      $TYPON
 5756  030356  112737  000001  030555  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 5757  030364  112737  000006  030557          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 5758  030372  112737  000005  030554  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 5759  030400  010346                          MOV     R3,-(SP)        ;;SAVE R3
 5760  030402  010446                          MOV     R4,-(SP)        ;;SAVE R4
 5761  030404  010546                          MOV     R5,-(SP)        ;;SAVE R5
 5762  030406  113704  030557                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 5763  030412  005404                          NEG     R4
 5764  030414  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 5765  030420  110437  030556                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 5766  030424  113704  030555                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 5767  030430  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 5768  030434  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
 5769  030436  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO 'C'
 5770  030440  000404                          BR      3$              ;;GO DO MSB
 5771  030442  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
 5772  030444  006105                          ROL     R5
```

```
5773  030446  006105                          ROL     R5
5774  030450  010503                          MOV     R5,R3
5775  030452  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
5776  030454  105337  030556                  DECB    $OMODE          ;;TYPE THIS DIGIT?
5777  030460  100016                          BPL     7$              ;;BR IF NO
5778  030462  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
5779  030466  001002                          BNE     4$              ;;TEST FOR 0
5780  030470  005704                          TST     R4              ;;SUPPRESS THIS 0?
5781  030472  001403                          BEQ     5$              ;;BR IF YES
5782  030474  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5783  030476  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5784  030502  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5785  030506  110337  030552                  MOVB    R3,8$           ;;SAVE FOR TYPING
5786  030512  104400  030552                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
5787  030516  105337  030554          7$:     DECB    $OCNT           ;;COUNT BY 1
5788  030522  003347                          BGT     2$              ;;BR IF MORE TO DO
5789  030524  002402                          BLT     6$              ;;BR IF DONE
5790  030526  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5791  030530  000744                          BR      2$              ;;GO DO THE LAST DIGIT
5792  030532  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
5793  030534  012604                          MOV     (SP)+,R4        ;;RESTORE R4
5794  030536  012603                          MOV     (SP)+,R3        ;;RESTORE R3
5795  030540  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5796  030546  012616                          MOV     (SP)+,(SP)
5797  030550  000002                          RTI                     ;;RETURN
5798  030552     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5799  030553     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5800  030554     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
5801  030555     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
5802  030556  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
5803
5804                                  ;;*****************************************************************
5805
5806                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
5807
5808                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
5809                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
5810                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
5811                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
5812                                  ;*REPLACED WITH SPACES.
5813                                  ;*CALL:
5814                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
5815                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
5816
5817  030560                          $TYPDS:
5818  030560  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
5819  030562  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
5820  030564  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
5821  030566  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
5822  030570  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
5823  030572  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
5824  030576  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
5825  030602  100004                          BPL     1$              ;;BR IF INPUT IS POS.
5826  030604  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
5827  030606  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
5828  030614  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
```

```
5829  030616  012703  030774                    MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
5830  030622  112723  000040                    MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
5831  030626  005002                    2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
5832  030630  016001  030764                    MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
5833  030634  160105                    3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
5834  030636  002402                            BLT     4$              ;;BR IF DONE
5835  030640  005202                            INC     R2              ;;INCREASE THE BCD DIGIT BY 1
5836  030642  000774                            BR      3$
5837  030644  060105                    4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
5838  030646  005702                            TST     R2              ;;CHECK IF BCD DIGIT=0
5839  030650  001002                            BNE     5$              ;;FALL THROUGH IF 0
5840  030652  105716                            TSTB    (SP)            ;;STILL DOING LEADING 0'S?
5841  030654  100407                            BMI     7$              ;;BR IF YES
5842  030656  106316                    5$:     ASLB    (SP)            ;;MSD?
5843  030660  103003                            BCC     6$              ;;BR IF NO
5844  030662  116663  000001  177777            MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
5845  030670  052702  000060            6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
5846  030674  052702  000040            7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
5847  030700  110223                            MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
5848  030702  005720                            TST     (R0)+           ;;JUST INCREMENTING
5849  030704  020027  000010                    CMP     R0,#10          ;;CHECK THE TABLE INDEX
5850  030710  002746                            BLT     2$              ;;GO DO THE NEXT DIGIT
5851  030712  003002                            BGT     8$              ;;GO TO EXIT
5852  030714  010502                            MOV     R5,R2           ;;GET THE LSD
5853  030716  000764                            BR      6$              ;;GO CHANGE TO ASCII
5854  030720  105726                    8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
5855  030722  100003                            BPL     9$              ;;BR IF NO
5856  030724  116663  177777  177776            MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
5857  030732  105013                    9$:     CLRB    (R3)            ;;SET THE TERMINATOR
5858  030734  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
5859  030736  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
5860  030740  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
5861  030742  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
5862  030744  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
5863  030746  104400  030774                    TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
5864  030752  016666  000002  000004            MOV     2(SP),4(SP)     ;;ADJUST THE STACK
5865  030760  012616                            MOV     (SP)+,(SP)
5866  030762  000002                            RTI                     ;;RETURN TO USER
5867  030764  023420                    $DTBL:  10000.
5868  030766  001750                            1000.
5869  030770  000144                            100.
5870  030772  000012                            10.
5871  030774  000004                    $DBLK:  .BLKW   4
5872
5873                                      ;;**************************************************************************
5874
5875                                      .SBTTL  TRAP DECODER
5876
5877                                      ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
5878                                      ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5879                                      ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5880                                      ;*GO TO THAT ROUTINE.
5881
5882  031004  010046                    $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
5883  031006  016600  000002                    MOV     2(SP),R0        ;;GET TRAP ADDRESS
5884  031012  005740                            TST     -(R0)           ;;BACKUP BY 2
```

```
 5885  031014  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 5886  031016  016000  031024                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 5887  031022  000200                          RTS     R0              ;;GO TO ROUTINE
 5888
 5889
 5890                                  .SBTTL  TRAP TABLE
 5891
 5892                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 5893                                  ;*BY THE "TRAP" INSTRUCTION.
 5894
 5895                                  ;       ROUTINE
 5896                                  ;       -------
 5897  031024                          $TRPAD:
 5898  031024  030116                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
 5899  031026  030356                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 5900  031030  030332                          $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 5901  031032  030372                          $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 5902  031034  030560                          $TYPDS  ;;CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
 5903  031036  030022                          $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
 5904  031040  030060                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
 5905
 5906  031042  032102                          CLEAN   ;;CALL=RSET     TRAP+16(104416) GO RESET ALL REGISTERS.
 5907  031044  032052                          ABORTT  ;;CALL=SKIPT    TRAP+20(104420) THIS WILL SKIP TO THE NEXT TEST
 5908  031046  032520                          MMDES   ;;CALL=MMSKIP   TRAP+22(104422) IF SWITCH # IS ON SKIP TO THE NEXT TEST
 5909  031050  032542                          MSIZER  ;;CALL=SIZE     TRAP+24(104424) DETERMINE THE HIGHEST ADDRESS IN MEMORY
 5910  031052  032172                          SKBADR  ;;CALL=SKPBAD   TRAP+26(104426) SKIP TEST IF ERROR ADDRESS REGISTER IS I
 5911  031054  032216                          SKBERR  ;;CALL=SKPBER   TRAP+30(104430) SKIP TEST IF ERROR REGISTER IS INOPERATI
 5912  031056  032234                          SKBCNR  ;;CALL=SKPBCN   TRAP+32(104432) SKIP TEST IF CONTROL REGISTER IS INOPERA
 5913  031060  032252                          SKBMNR  ;;CALL=SKPBMN   TRAP+34(104434) SKIP TEST IF MAINTENANCE REGISTER IS INO
 5914  031062  032270                          SKBHMR  ;;CALL=SKPBHM   TRAP+36(104436) SKIP TEST IF HIT/MISS REGISTER IS IN OPE
 5915
 5916                                  ;;**************************************************************************
 5917
 5918                                  .SBTTL  POWER DOWN AND UP ROUTINES
 5919
 5920                                  ;POWER DOWN ROUTINE
 5921  031064  012737  031212  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 5922  031072  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
 5923  031100  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 5924  031102  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5925  031104  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 5926  031106  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 5927  031110  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 5928  031112  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 5929  031114  010637  031216                  MOV     SP,$SAVR6       ;;SAVE SP
 5930  031120  012737  031132  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 5931  031126  000000                          HALT
 5932  031130  000776                          BR      .-2             ;;HANG UP
 5933
 5934                                  ;POWER UP ROUTINE
 5935  031132  013706  031216          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
 5936  031136  005037  031216                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 5937  031142  005237  031216          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
 5938  031146  001375                          BNE     1$              ;;OF  WORD
 5939  031150  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 5940  031152  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
```

```
5941  031154  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
5942  031156  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
5943  031160  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
5944  031162  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5945  031164  012737  031064  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
5946  031172  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
5947  031200  104400                          TYPE                    ;;REPORT THE POWER FAILURE
5948  031202  033373                $PWRMG:   .WORD   POWERM          ;;POWER FAIL MESSAGE POINTER
5949  031204  012716                          MOV     (PC)+,(SP)      ;;RESTART AT START
5950  031206  003014                $PWRAD:   .WORD   START           ;;RESTART ADDRESS
5951  031210  000002                          RTI
5952  031212  000000                $ILLUP:   HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
5953  031214  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
5954  031216  000000                $SAVR6:   0                       ;;PUT THE SP HERE
5955                                ;;*****************************************************************************
5956
5957                                .SBTTL  ROUTINE TO SIZE MEMORY
5958
5959                                ;*CALL:
5960                                ;*      JSR     PC,$SIZE
5961                                ;*      RETURN
5962                                ;*$LSTAD WILL CONTAIN:
5963                                ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
5964                                ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
5965                                ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
5966                                ;*$KT11 IS THE MEMORY MANAGEMENT KEY
5967                                ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
5968                                ;*      MUST BE SETUP BEFORE THE CALL
5969                                ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
5970                                ;*      DETERMINED BY ROUTINE
5971                                ;* --NOTE--
5972                                ;*THIS ROUTINE SUPPORTS PDP 11/74.
5973                                ;*IF ACTUAL MEMORY IS LESS THAN THAT INDICATED BY THE SIZE REGISTER
5974                                ;*AND A REFERENCE IS MADE TO A MEMORY ADDRESS THAT IS GREATER THAN
5975                                ;*ACTUAL MEMORY BUT LESS THAN SIZE REGISTER ((INDICATED), THEN A
5976                                ;*MEMORY REFERENCE TIMEOUT TO VECTOR 114 WILL OCCUR.
5977
5978  031220  010046                $SIZE:    MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
5979  031222  010146                          MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
5980  031224  010246                          MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
5981  031226  010346                          MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
5982  031230  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
5983  031234  013746  000006                  MOV     @#ERRVEC+2,-(SP)
5984  031240  013746  000114                  MOV     @#114,-(SP)     ;;SAVE PRESENT PARITY VECOT PS & PC
5985  031244  013746  000116                  MOV     @#116,-(SP)
5986  031250  010600                          MOV     SP,R0           ;;SAVE THE STACK POINTER
5987  031252  013737  177776  000006          MOV     @#PS,@#ERRVEC+2 ;;SET ERRVEC PS TO PRESENT PS
5988  031260  012701  003776                  MOV     #3776,R1        ;;SETUP ADDRESS
5989  031264  105727                          TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
5990  031266  000200                $KT11:    .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
5991  031270  100065                          BPL     $CORE           ;;BR IF NO
5992  031272  012737  031436  000004          MOV     #$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
5993  031300  005737  177572                  TST     @#SR0           ;;KT11 ARE YOU THERE?
5994  031304  052737  100000  031266          BIS     #100000,$KT11   ;;YES--SET KT11 KEY
5995  031312  005046                          CLR     -(SP)           ;;INITIALIZE FOR 'PAR' LOADING
5996  031314  012702  172340                  MOV     #KIPAR0,R2      ;;ADDRESS OF FIRST 'PAR'
```

```
 5997  031320  012703  000010                    MOV     #^D8,R3           ;;LOAD EIGHT 'PAR.'S'' AND EIGHT 'PDR.'S''
 5998  031324  012762  077406  177740  1$:       MOV     #77406,-40(R2)    ;;PDR = 4K, UP, READ/WRITE
 5999  031332  011622                            MOV     (SP),(R2)+        ;;LOAD 'PAR''
 6000  031334  062716  000200                    ADD     #200,(SP)         ;;UPDATE FOR NEXT 'PAR''
 6001  031340  077307                            SOB     R3,1$             ;;LOOP UNTIL ALL EIGHT ARE LOADED
 6002  031342  012742  177600                    MOV     #177600,-(R2)     ;;SETUP KIPAR7 FOR I/O
 6003  031346  005042                            CLR     -(R2)             ;;SETUP KIPAR6 FOR TESTING
 6004  031350  012737  031366  000004            MOV     #2$,@#ERRVEC      ;;CATCH TIMEOUT IF NO SR3
 6005  031356  012737  000020  172516            MOV     #20,@#SR3         ;;ENABLE 22-BIT ADDRESSING
 6006  031364  000401                            BR      3$                ;;THIS PDP-11 HAS A SR3 REG.
 6007  031366  022626                    2$:     CMP     (SP)+,(SP)+       ;;CLEAN OFF THE STACK--NO SR3.
 6008  031370  005237  177572            3$:     INC     @#SR0             ;;TURN ON MEMORY MANAGEMENT
 6009  031374  012737  031426  000004            MOV     #$KTOUT,@#ERRVEC  ;;SET FOR TIME OUT
 6010  031402  012737  031550  000114            MOV     #$MTMOUT,@#114    ;;SET FOR MEM REF TIMEOUT
 6011  031410  005737  143776            4$:     TST     @#143776          ;;TRAP ON NON-EX-MEM
 6012  031414  062712  000040                    ADD     #40,(R2)          ;;MAKE A 1K STEP
 6013  031420  023712  172356                    CMP     @#KIPAR7,(R2)     ;;LAST ONE?
 6014  031424  101371                            BHI     4$                ;;NO--TRY IT
 6015  031426  011202                    $KTOUT: MOV     (R2),R2           ;;GET LAST BANK+1
 6016  031430  005037  177572                    CLR     @#SR0             ;;TURN OFF MEMORY MANAGEMENT
 6017  031434  000421                            BR      $SIZEX
 6018  031436  042737  100000  031266  $KTNEX: BIC     #100000,$KT11     ;;KT11 NON-EXISTENT
 6019  031444  012737  031474  000004  $CORE:  MOV     #$CROUT,@#ERRVEC  ;;SET FOR TIMEOUT
 6020  031452  005002                            CLR     R2                ;;SET UP BANK
 6021  031454  062701  004000            1$:     ADD     #4000,R1          ;;INCREMENT BY 1K
 6022  031460  062702  000040                    ADD     #40,R2            ;;1K STEP
 6023  031464  005711                            TST     (R1)              ;;TRAP ON TIME OUT
 6024  031466  022701  177776                    CMP     #177776,R1        ;;LAST ONE
 6025  031472  001370                            BNE     1$                ;;NO--TRY AGAIN
 6026  031474  162701  004000            $CROUT: SUB     #4000,R1
 6027  031500  162702  000040            $SIZEX: SUB     #40,R2            ;;DROP BACK
 6028  031504  010006                            MOV     R0,SP             ;;RESTORE THE STACK
 6029  031506  012637  000116                    MOV     (SP)+,@#116       ;;RESTOR PARITY VECTOR
 6030  031512  012637  000114                    MOV     (SP)+,@#114
 6031  031516  012637  000006                    MOV     (SP)+,@#ERRVEC+2  ;;RESTORE ERROR VECTOR
 6032  031522  012637  000004                    MOV     (SP)+,@#ERRVEC
 6033  031526  010137  031602                    MOV     R1,$LSTAD         ;;LAST ADDRESS
 6034  031532  010237  031604                    MOV     R2,$LSTBK         ;;LAST BANK
 6035  031536  012603                            MOV     (SP)+,R3          ;;RESTORE R3
 6036  031540  012602                            MOV     (SP)+,R2          ;;RESTORE R2
 6037  031542  012601                            MOV     (SP)+,R1          ;;RESTORE R1
 6038  031544  012600                            MOV     (SP)+,R0          ;;RESTORE R0
 6039  031546  000207                            RTS     PC
 6040  031550  032737  000001  177744  $MTMOUT: BIT     #BIT0,@#MEMERR    ;;MAKE SURE TRAP TO 114 IS DUE
 6041  031556  001005                            BNE     1$                ;;TO MEMORY REFERENCE TIMEOUT
 6042                                                                      ;;IF NOT, IS IT AN ABORT?
 6043  031560  032737  100000  177744            BIT     #BIT15,@#MEMERR   ;;CPU ABORT?
 6044  031566  001001                            BNE     1$                ;;IF YES, EXIT OUT
 6045  031570  000002                            RTI                       ;;IF NOT, CONTINUE
 6046  031572  012737  177777  177744  1$:      MOV     #-1,@#MEMERR      ;;CLEAR THE MEM ERROR REG
 6047  031600  000712                            BR      $KTOUT
 6048  031602  000000                    $LSTAD: .WORD   0                 ;;CONTAINS THE LAST ADDRESS
 6049  031604  000000                    $LSTBK: .WORD   0                 ;;CONTAINS THE LAST BANK
 6050
 6051
 6052                                    ;;*****************************************************************************
```

```
                                .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

                                ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
                                ;*UNSIGNED OCTAL ASCIZ NUMBER.
                                ;*CALL
                                ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                                ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
                                ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK

031606  104412          $DB20:  SAVREG                          ;;SAVE ALL REGISTERS
031610  016601  000002          MOV     2(SP),R1                ;;PICKUP THE POINTER TO LOW WORD
031614  012705  031725          MOV     #$OCTVL+13.,R5          ;;POINTER TO DATA TABLE
031620  012704  000014          MOV     #12.,R4                 ;;DO ELEVEN CHARACTERS
031624  012703  177770          MOV     #^C7,R3                 ;;MASK
031630  012100                  MOV     (R1)+,R0                ;;LOWER WORD
031632  012101                  MOV     (R1)+,R1                ;;HIGH WORD
031634  005002                  CLR     R2                      ;;TERMINATOR
031636  110245          1$:     MOVB    R2,-(R5)                ;;PUT CHARACTER IN DATA TABLE
031640  010002                  MOV     R0,R2                   ;;GET THIS DIGIT
031642  005304                  DEC     R4                      ;;COUNT THIS CHARACTER
031644  003007                  BGT     3$                      ;;BR IF NOT THE LAST DIGIT
031646  001405                  BEQ     2$                      ;;BR IF IT IS THE LAST DIGIT
031650  005205                  INC     R5                      ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
031652  010566  000002          MOV     R5,2(SP)                ;;ASCIZ CHAR. & PUT IT ON THE STACK
031656  104414                  RESREG                          ;;RESTORE ALL REGISTERS
031660  000207                  RTS     PC                      ;;RETURN TO USER
031662  006203          2$:     ASR     R3                      ;;POSITION THE MASK FOR THE LAST DIGIT
031664  006001          3$:     ROR     R1                      ;;POSITION THE BINARY NUMBER FOR
031666  006000                  ROR     R0                      ;;      THE NEXT OCTAL DIGIT
031670  006001                  ROR     R1
031672  006000                  ROR     R0
031674  006001                  ROR     R1
031676  006000                  ROR     R0
031700  040302                  BIC     R3,R2                   ;;MASK OUT ALL JUNK
031702  062702  000060          ADD     #'0,R2                  ;;MAKE THIS CHAR. ASCII
031706  000753                  BR      1$                      ;;GO PUT IT IN THE DATA TABLE
031710  000016          $OCTVL: .BLKB   14.                     ;;RESERVE DATA TABLE

                                ;THIS ROUTINE IS CALLED BY UNEXPECTED TRAPS TO VECTOR ERRVEC.
                                ;THE ERROR IS REPORTED AND CONTROL IS TRANSFERRED BACK TO THE TEST
                                ;FOLLOWING THE ONE THAT WAS INTERRUPTED WHEN THE ERROR OCCURRED!
031726  011637  001226  CPSPUR: MOV     (SP),$TMP1
031732  012737  031750  001230          MOV     #1$,$TMP2
031740  013737  177766  001232          MOV     @#CPUERR,$TMP3
031746  022626                  CMP     (SP)+,(SP)+             ;RESET THE STACK
031750  104150          1$:     ERROR   150
031752  104420                  SKIPT

                                ;THIS ROUTINE HANDLE UNEXPECTED TRAPS TO #CACHVEC.
031754  012737  032044  000114  SPUR:   MOV     #10$,@#CACHVEC
031762  013700  177744          MOV     @#MEMERR,R0
031766  032700  000014          BIT     #14,R0                  ;SEE IF IT WAS A MAIN MEMORY PARITY ERROR.
031772  001403                  BEQ     9$
031774  013700  177740          MOV     @#LOADRS,R0             ;IF IT WAS THEN THE BAD PARITY IS
```

```
                                        TST     (R0)            ;CACHED AND MUST BE PURGED!!!!!
032002  012737  031754  000114  9$:     MOV     #SPUR,@#CACHVEC
032010  013737  177744  001234          MOV     @#MEMERR,$TMP4  ;TRAP HERE IF AN UNEXPECTED
032016  013737  177740  001226          MOV     @#LOADRS,$TMP1  ;ERROR, PARITY, OCCURS.
032024  013737  177742  001230          MOV     @#HIADRS,$TMP2
032032  011637  001232                  MOV     (SP),$TMP3
032036  022626                          CMP     (SP)+,(SP)+
032040  104014                  1$:     ERROR   14
032042  104420                          SKIPT                   ;?????
032044  022626                  10$:    CMP     (SP)+,(SP)+
032046  000137  032002                  JMP     9$

                                ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL SKIPT.
                                ;IT TELLS THE USER THAT THE CURRENT TEST HAS BEEN
                                ;ABORTED AND THAT CONTROL IS BEING PASSED TO THE NEXT TEST.
032052  011637  001226          ABORTT: MOV     (SP),$TMP1
032056  112737  000015  001114          MOVB    #15,$ITEMB
032064  022626                          CMP     (SP)+,(SP)+
032066  004737  032616                  JSR     PC,ERTYPE
032072  104416                          RSET
032074  000177  000000                  JMP     @SKAD           ;GO TO @SKAD, WHICH SHOULD
                                                                ;BE SET TO THE
032100  000000                  SKAD:   .WORD   0               ;ADDRESS OF THE NEXT TEST.


                                ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL RSET. IT CLEARS ALL
                                ;THE IMPORTANE REGISTERS AND RESETS THE STACK.
032102                          CLEAN:

032102  012737  031754  000114          MOV     #SPUR,@#CACHVEC
032110  012737  031726  000004          MOV     #CPSPUR,@#ERRVEC
032116  011637  032170                  MOV     (SP),BACKAD
032122  012706  001100                  MOV     #STACK,SP
032126  005037  177750                  CLR     @#MAINT         ;CLEAR ALL CONTROL AND ERROR
032132  005037  177572                  CLR     @#MMR0          ;REGISTERS.
032136  005037  172516                  CLR     @#MMR3
032142  005037  177746                  CLR     @#CONTRL
032146  012737  177777  177744          MOV     #-1,@#MEMERR
032154  005037  177766                  CLR     @#CPUERR
032160  005037  177776                  CLR     @#PSW
032164  000177  000000                  JMP     @BACKAD
032170  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
                                ;       SKPBMN          SKIPT IF BAD MAINTENANCE REGISTER
                                ;       SKPBHM          SKIPT IF BAD HIT/MISS REGISTER
                                ;


032172  005737  032310          SKBADR: TST     LOAFLG
```

```
 6165  032176  001004                        BNE     1$
 6166  032200  005737  032312                TST     HIAFLG
 6167  032204  001001                        BNE     1$
 6168  032206  000002                        RTI
 6169  032210  104400                1$:     TYPE
 6170  032212  034355                        .WORD   ADRNG
 6171  032214  000433                        BR      SKRNG
 6172
 6173  032216  005737  032314        SKBERR: TST     MMRFLG
 6174  032222  001001                        BNE     1$
 6175  032224  000002                        RTI
 6176  032226  104400                1$:     TYPE
 6177  032230  034465                        .WORD   ERRNG
 6178  032232  000424                        BR      SKRNG
 6179
 6180  032234  005737  032316        SKBCNR: TST     CONFLG
 6181  032240  001001                        BNE     1$
 6182  032242  000002                        RTI
 6183  032244  104400                1$:     TYPE
 6184  032246  034565                        .WORD   CNRNG
 6185  032250  000415                        BR      SKRNG
 6186
 6187  032252  005737  032320        SKBMNR: TST     MANFLG
 6188  032256  001001                        BNE     1$
 6189  032260  000002                        RTI
 6190  032262  104400                1$:     TYPE
 6191  032264  034667                        .WORD   MNRNG
 6192  032266  000406                        BR      SKRNG
 6193
 6194  032270  005737  032322        SKBHMR: TST     HIMFLG
 6195  032274  001001                        BNE     1$
 6196  032276  000002                        RTI
 6197  032300  104400                1$:     TYPE
 6198  032302  034775                        .WORD   HMRNG
 6199
 6200  032304  022626                SKRNG:  CMP     (SP)+,(SP)+     ;RESET THE STACK AND GO TO THE
 6201  032306  104420                        SKIPT                   ;NEXT TEST!!!!!
 6202
 6203  032310  000000                LOAFLG: .WORD   0               ;THESE ARE FLAGS USED TO DESIGNATE
 6204  032312  000000                HIAFLG: .WORD   0               ;EITHER A GOOD OR A BAD REGISTER.
 6205  032314  000000                MMRFLG: .WORD   0               ;GOOD WILL BE DESIGNATED BY A
 6206  032316  000000                CONFLG: .WORD   0               ;0 BAD BY A NOT ZERO!!
 6207  032320  000000                MANFLG: .WORD   0
 6208  032322  000000                HIMFLG: .WORD   0
 6209  032324  000000                LOAFL2: .WORD   0
 6210  032326  000000                HIAFL2: .WORD   0
 6211  032330  000000                MMRFL2: .WORD   0
 6212  032332  000000                CONFL2: .WORD   0
 6213  032334  000000                MANFL2: .WORD   0
 6214  032336  000000                HIMFL2: .WORD   0
 6215
 6216                                ;THIS ROUTINE IS CALLED TO DETERMINE THE PARITY OF
 6217                                ;A DATA PATTERN. THE PATTERN WHICH IS TAKEN BY THIS
 6218                                ;ROUTINE AS ITS ARGUMENT SHOULD BE PUT IN R0. THEN
 6219                                ;TRANSFER CONTROL HERE BY EXECUTING:
 6220                                ;       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.
032340  012701  000001          PARCNT: MOV     #1,R1
032344  005002                          CLR     R2
032346  030100                  1$:     BIT     R1,R0
032350  001401                          BEQ     2$
032352  005202                          INC     R2
032354  006301                  2$:     ASL     R1
032356  103373                          BCC     1$
032360  000207                          RTS     PC

                                        ;THIS ROUTINE IS CALLED TO RESTORE THE TOP 1500 (DEC) WORDS IN THE
                                        ;FIRST 28K OF MEMORY. THIS SHOULD EFFECTIVELY RESTORE ANY MONITOR
                                        ;OR LOADER THAT WAS PRESENT BEFORE THIS PROGRAM BEGAN EXECUTION.
                                        ;CONTROL IS PASSED TO THIS ROUTINE BY AN INTERRUPT FROM THE TTY KEYBOARD
                                        ;WHEN ANY CHARACTER IS TYPED ON THE KEYBOARD. IF THE CHARACTER.
                                        ;TURNS OUT TO BE A ^C (CONTROL-C) THEN MEMORY IS RESTORED. IF THE
                                        ;CHARACTER IS NOT ^C THEN A RETURN IS MADE TO THE TEST FOLLOWING
                                        ;THE ONE WHOSE EXECUTION WAS INTERRUPTED BY THE KEYBOARD INTERRUPT.
032362  005037  177750          RESMON: CLR     @#MAINT
032366  017700  146546                  MOV     @$TKB,R0
032372  104416                          RSET
032374  005003                          CLR     R3
032376  042700  000200                  BIC     #BIT7,R0        ;GET THE CHARACTER, INITIALIZE THE REGISTERS
032402  022700  000003                  CMP     #3,R0           ;AND SEE IF THE CHARACTER WAS ^C.
032406  001032                          BNE     NOCNC           ;BRANCH AND GO TO NEXT TEST IF NOT.
032410  104400                          TYPE                    ;ECHOE THE CONTROL-C AS '^C'
032412  033330                          .WORD   CONCMS
032414  012704  002734          CHAINQ: MOV     #^D1500,R4              ;AND RESTORE THE MONITOR.
032420  012701  052700                  MOV     #BOTTOM+4,R1
032424  012702  160000                  MOV     #160000,R2
032430  012142                  1$:     MOV     (R1)+,-(R2)
032432  077402                          SOB     R4,1$
032434  012737  177777  032516        MOV     #-1,MONF        ;RESET THE MONITOR RESTORED FLAG.
032442  020327  125252                  CMP     R3,#125252      ;SEE IF THE MONITOR IS BEING RESTORED
                                                                ;BY THE .$EOP ROUTINE.
032446  001001                          BNE     STOP            ;IF NOT GO HALT, OTHERWISE RETURN TO .$EOP
032450  000207                          RTS     PC
032452  104400                  STOP:   TYPE                    ;TYPE THE MONITOR RESTORED MESSAGE.
032454  033334                          .WORD   MMESRS
032456  013737  032514  000060        MOV     MONTTY,@#TKVEC
032464  000000                          HALT                    ;AND HALT!!
032466  012737  032362  000060        MOV     #RESMON,@#TKVEC
032474  005077  146440          NOCNC:  CLR     @$TKB           ;NOT CONTROL C SO RETURN TO NEXT TEST.
032500  152777  000100  146430        BISB    #BIT6,@$TKS
032506  104416                          RSET
032510  000177  177364                  JMP     @SKAD           ;RETURN.
032514  000000                  MONTTY: .WORD   0               ;TEMPORARY STORAGE FOR THE INITIAL
                                                                ;CONTENTS OF THE TTY KEYBOARD INTERRUPT VECTOR.
032516  177777                  MONF:   .WORD   177777          ;FLAG, IF NOT -1 THE MONITOR IS SAVED!!


                                        ;THIS ROUTINE IS CALLED BY THE TRAP CALL MMSKIP. IT LOOKS
                                        ;AT THE SWITCH REGISTER AND DETERMINES WHETHER OR NOT
                                        ;SWITCH #7 IS ON. IF SO THE CURRENT TEST IS SKIPPED
```

```
                                ;AND THE NEXT TEST IS ENTERED. A SSKAD MUST BE ISSUED
                                ;BEFORE THE MMSKIP.
                                ;THE PURPOSE OF SWITCH #7 IS TO CAUSE THE DELETION OF THE
                                ;EXECUTION OF ANY TEST WHICH RELIES ON MEMORY MANAGEMENT
                                ;FOR ITS OPERATION.

032520  032737  000200  177570  MMDES:  BIT     #SW7,@#SWR
032526  001001                          BNE     1$              ;IS THE SWITCH ON?
032530  000002                          RTI                     ;NO, SO RETURN.
032532  022626                  1$:     CMP     (SP)+,(SP)+
032534  104416                          RSET
032536  000177  177336                  JMP     @SKAD           ;YES, GO TO THE NEXT TEST.
                                ;THIS ROUTINE IS CALLED TO DETERMINE THE HIGHEST POSSIBLE
                                ;ADDRESS IN MEMORY. IT IS CALLED THUS, BY TRAP CALL SIZE:
                                ;               SIZE
                                ;       LOORDA: .WORD   0
                                ;       HIORDA: .WORD   0
                                ;       NXTINST:
                                ;THE LOW ORDER 16-BITS OF THE ADDRESS ARE LEFT IN THE
                                ;WORD DIRECTLY FOLLOWING THE CALL. THE HIGH ORDER 6-BITS
                                ;ARE LEFT IN THE NEXT WORD AND CONTROL IS RETURNED
                                ;TO THE THIRD WORD FOLLOWING THE CALL.
032542  010046                  MSIZER: MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0 AND R1
032544  010146                          MOV     R1,-(SP)        ;GET THE ADDRESS OF
032546  016600  000004                  MOV     4(SP),R0        ;THE CALL OF THE STACK.
032552  013710  177760                  MOV     @#SIZELO,(R0)
032556  005060  000002                  CLR     2(R0)
032562  012701  000006                  MOV     #6,R1           ;ROTATE THE 16-BIT 'BLOCK'
                                                                ;NUMBER 6-BITS TO THE
032566  006310                  1$:     ASL     (R0)            ;LEFT AND TURN ON LOW ORDER
032570  006160  000002                  ROL     2(R0)           ;BITS 1-5 LEAVING BIT-0
032574  077104                          SOB     R1,1$           ;OFF SO AS TO CREATE
032576  052710  000076                  BIS     #76,(R0)        ;THE 22-BIT PHYSICAL ADDRESS OF
                                                                ;THE HIGHEST WORD IN
                                                                ;MEMORY.
032602  022020                          CMP     (R0)+,(R0)+     ;DETERMINE THE RETURN ADDRESS

032604  010066  000004                  MOV     R0,4(SP)        ;AND LEAVE ON THE STACK FOR
                                                                ;AN RTI.
032610  012601                          MOV     (SP)+,R1        ;RESTORE R1 AND R0.
032612  012600                          MOV     (SP)+,R0
032614  000002                          RTI                     ;RETURN
                                ;THIS ROUTINE IS USED TO TYPE AN ERROR MESSAGE
                                ;WHICH IS IN THE DATA TABLE. IT IS CALLED BY
                                ;THE $ERROR ROUTINE OR BY FIRST SETTING THE $ITEMB
                                ;BYTE EQUAL TO THE ERROR TABLE ITEM NUMBER THAT IS
                                ;TO BE PRINTED OUT AND THEN EXECUTING A JSR PC,ERTYPE
032616  104400                  ERTYPE: TYPE
032620  001305                          .WORD   $CRLF

032622  010046                          MOV     R0,-(SP)        ;SAVE R0
032624  005000                          CLR     R0

032626  113700  001114                  MOVB    $ITEMB,R0       ;GET THE ITEM NUMBER
032632  001005                          BNE     1$              ;ZERO?
032634  013746  001116                  MOV     $ERRPC,-(SP)    ;YES. TYPE JUST THE PC
```

```
                                        TYPOC                   ;OF THE ERROR CALL.
                                        JMP     ERT5
                
                                1$:     DEC     R0              ;MAKE R0 AN INDEX FOR THE
                                        ASH     #3,R0           ;ERROR TABLE
                                        ADD     #$ERRTB,R0
                                        MOV     (R0)+,2$        ;TYPE EM, ERROR MESSAGE.
                                        BEQ     3$
                                        TYPE
                                2$:     .WORD   0
                                        TYPE
                                        .WORD   $CRLF
                                3$:     MOV     (R0)+,4$        ;TYPE DH, DATA HEADER
                                        BEQ     5$
                                        TYPE
                                4$:     .WORD   0
                                        TYPE
                                        .WORD   $CRLF
                                5$:     MOV     R1,-(SP)        ;SAVE R1
                                        MOV     (R0)+,R1        ;GET DT, DATA TABLE ADDRESS
                                        BNE     6$
                                        JMP     ERT4            ;JMP IF NO ERROR TABLE.
                                6$:     MOV     (R0)+,R0        ;GET DF, DATA FORMAT ADDRESS
                                ERT1:   TSTB    (R0)            ;DATA FORMAT ENTRY EQUALS
                                        BNE     7$              ;ZERO?
                                        MOV     @(R1)+,-(SP)    ;YES, SO TYPE A 16-BIT
                                        TYPOC                   ;OCTAL NUMBER
                                        BR      ERT2
                                7$:     CMPB    #1,(R0)         ;FORMAT EQUALS 1?
                                        BNE     8$
                                        MOV     @(R1)+,-(SP)    ;YES, TYPE A DECIMAL NUMBER
                                        TYPDS
                                        BR      ERT2
                
                                8$:     CMPB    #2,(R0)         ;FORMAT 2?
                                        BNE     9$
                                85$:    MOV     (R1)+,-(SP)     ;YES, TYPE A 22-BIT NUMBR
                                        JSR     PC,$DB2O        ;CALL $DB2O TO CONVERT THE
                                        ADD     #3,(SP)         ;BINARY TO ASCII
                                        MOV     (SP)+,29$       ;TYPE THE STRING
                                        TYPE
                                29$:    .WORD   0
                                        BR      ERT2
                
                                9$:     CMPB    #4,(R0)         ;FORMAT 4?
                                        BNE     10$
                                        MOV     @(R1)+,-(SP)    ;YES, TYPE A 16-BIT
                                        TYPOS                   ;OCTAL NUMBER SUPRESSING
                                        .BYTE   16              ;LEADING ZEROES
                                        .BYTE   0
                                        BR      ERT2
                                10$:    CMPB    #3,(R0)         ;FORMAT 3?
                                        BNE     11$
                                        MOV     @(R1)+,-(SP)    ;YES CONVERT 16-BIT
                                        MOV     #-1,TVADFL      ;VIRTUAL ADDRESS TO 32-BIT
                                        JSR     PC,TYPVAD       ;PHYSICAL ADDRESS AND TYPE
```

```
6389  033050  000434                        BR      ERT2            ;RELOCATE ONLY IF SEG. IS ON!
6390  033052  122710  000005        11$:    CMPB    #5,(R0)         ;FORMAT 5?
6391  033056  001005                        BNE     12$
6392  033060  012137  033066              MOV     (R1)+,20$       ;PRINT ASCIZ STRING
6393  033064  104400                        TYPE
6394  033066  000000        20$:    .WORD   0
6395  033070  000426                        BR      ERT3
6396
6397  033072  122710  000006        12$:    CMPB    #6,(R0)         ;FORMAT 6
6398  033076  001005                        BNE     13$
6399  033100  005037  033164              CLR     TVADFL
6400  033104  004737  033172              JSR     PC,TYPVAD
6401  033110  000414                        BR      ERT2
6402
6403  033112  122710  000007        13$:    CMPB    #7,(R0)         ;FORMAT 7?
6404  033116  001010                        BNE     14$
6405  033120  012146                        MOV     (R1)+,-(SP)
6406  033122  004737  031606              JSR     PC,$DB20
6407  033126  012637  033134              MOV     (SP)+,45$
6408  033132  104400                        TYPE
6409  033134  000000        45$:    .WORD   0
6410  033136  000401                        BR      ERT2
6411
6412  033140  000000        14$:    HALT                            ;?????
6413
6414  033142  104400        ERT2:   TYPE                            ;PRINT A TAB AFTER TYPING AN
6415  033144  033440                .WORD   $TAB                    ;ERROR TABLE ENTRY OF ALL MODES
6416                                                                ;EXCEPT ASCIZ
6417  033146  005200        ERT3:   INC     R0                      ;POINT TO THE NEXT FORMAT BYTE
6418  033150  005711                TST     (R1)                    ;IS THERE ANOTHER ENTRY?
6419  033152  001401                BEQ     ERT4
6420  033154  000665                BR      ERT1                    ;YES, PROCESS IT
6421                                                                ;OTHERWISE:
6422  033156  012601        ERT4:   MOV     (SP)+,R1                ;RESTORE R1
6423  033160  012600        ERT5:   MOV     (SP)+,R0                ;RESTORE R0
6424  033162  000207                RTS     PC                      ;AND RETURN
6425
6426  033164  000000        TVADFL: .WORD   0                       ;FLAG USED TO TELL TYVAD
6427                                                                ;WHETHER TO CONDITIONALLY
6428                                                                ;OR UNCONDITIONALLY RELOCATE
6429                                                                ;WHEN TYPING AN ADDRESS.
6430                                                                ;-1 OR 0 RESPECTIVELY
6431
6432  033166  000000        TVADLO: .WORD   0                       ;REGISTERS FOR THE 22-BIT
6433  033170  000000        TVADHI: .WORD   0                       ;ADDRESS COMPUTED BY TYVAD.
6434
6435                        ;ROUTINE WHICH CONVERTS A 16-BIT ADDRESS TO A 22-BIT
6436                        ;ADDRESS. IF TVADFL IS -1, THEN CONVERT TO THE 22-BIT
6437                        ;REAL ADDRESS DEPENDENT ON SEG BEING ON OR OFF FOR RELOCATION.
6438                        ;IF TVADFL IS ZERO THEN UNCONDITIONAL USE THE KERNAL
6439                        ;PAR WHICH IS APPROPIATE TO DO RELOCATION.
6440  033172  104412        TYPVAD: SAVREG
6441  033174  016601  000002       MOV     2(SP),R1                ;GET THE VIRTUAL
6442  033200  010137  033166       MOV     R1,TVADLO               ;ADDRESS
6443  033204  005037  033170       CLR     TVADHI
6444  033210  005737  033164       TST     TVADFL                  ;CONDITIONALLY RELOCATE?
```

```
6445  033214  001404                        BEQ     1$
6446  033216  032737  000001  177572        BIT     #1,@#MMR0       ;YES, SEE IF MEMORY
6447  033224  001424                        BEQ     2$              ;MANAGEMENT IS ON
6448  033226  005000                1$:     CLR     R0              ;RELOCATE
6449  033230  073027  000003                ASHC    #3,R0           ;LEFT SHIFT R0 AND R1
6450  033234  006300                        ASL     R0              ;THREE PLACES. R0 ONE
6451                                                                ;MORE SO THAT IT CONTAINS
6452                                                                ;2 X THE UPPER 3-BITS OF
6453  033236  000241                        CLC                     ;THE VIRTUAL ADDRESS
6454  033240  006001                        ROR     R1              ;RESTORE R1 TO THE OFFSET
6455  033242  006001                        ROR     R1              ;OF THE VIRTUAL ADDRESS
6456  033244  006001                        ROR     R1              ;TO THE PAR
6457  033246  062700  172340                ADD     #KIPAR0,R0      ;DETERMINE THE CORRECT PAR'S
6458                                                                ;ADDRESS
6459  033252  011003                        MOV     (R0),R3         ;GET ITS CONTENTS
6460  033254  005002                        CLR     R2
6461  033256  073227  000006                ASHC    #6,R2           ;MAKE THE BLOCK COUNT
6462                                                                ;A 22-BIT ADDRESS.
6463  033262  060103                        ADD     R1,R3           ;ADD THE OFFSET TO THE
6464  033264  005502                        ADC     R2              ;BASE ADDRESS
6465
6466  033266  010237  033170                MOV     R2,TVADHI
6467  033272  010337  033166                MOV     R3,TVADLO
6468  033276  012746  033166        2$:     MOV     #TVADLO,-(SP)   ;CALL $DB2O TO CONVERT THE
6469  033302  004737  031606                JSR     PC,$DB2O        ;22-BIT
6470  033306  062716  000003                ADD     #3,(SP)         ;TYPE ONLY 8 DIGITS.
6471  033312  012637  033320                MOV     (SP)+,3$
6472  033316  104400                        TYPE
6473  033320  000000                3$:     .WORD   0
6474  033322  104414                        RESREG                  ;RESTORE THE REGISTERS
6475  033324  012616                        MOV     (SP)+,(SP)      ;LEAVE ONLY THE RETURN
6476                                                                ;ADDRESS ON THE STACK.
6477  033326  000207                        RTS     PC              ;RETURN
6478
6479                                        ;SPECIAL MESSAGES:
6480
6481  033330  041536  000200               CONCMS: .ASCIZ  '^C'<CRLF>
6482
6483  033334  047515  044516  047524       MMESRS: .ASCIZ  'MONITOR (OR LOADER) RESTORED!'<CRLF>
6484  033342  020122  047450  020122
6485  033350  047514  042101  051105
6486  033356  020051  042522  052123
6487  033364  051117  042105  100041
6488  033372     000
6489
6490  033373     200  047520  042527       POWERM: .ASCIZ  <CRLF>'POWER FAILURE, PROGRAM RESTARTING'<CRLF><CRLF>
6491  033400  020122  040506  046111
6492  033406  051125  026105  050040
6493  033414  047522  051107  046501
6494  033422  051040  051505  040524
6495  033430  052122  047111  100107
6496  033436  000200
6497
6498  033440  000011                $TAB:   .ASCIZ  <TAB>
6499
6500  033442  042600  050130  041505       MTA5:   .ASCII  <CRLF>'EXPECTED DATA:'<CRLF>
```

```
 6501  033450  042524  020104  040504
 6502  033456  040524  100072
 6503  033462  051107  052517  020120          .ASCIZ  'GROUP 0.GROUP 1.MEM EV.'<TAB>'MEM ODD.'<CRLF>
 6504  033470  027060  051107  052517
 6505  033476  020120  027061  042515
 6506  033504  020115  053105  004456
 6507  033512  042515  020115  042117
 6508  033520  027104  000200
 6509
 6510  033524  042200  052101  020101  MTA11:  .ASCII  <CRLF>'DATA WRITTEN.'<TAB>'TEST ADDR.'<TAB>'ERROR REG.'<CRLF>
 6511  033532  051127  052111  042524
 6512  033540  027116  052011  051505
 6513  033546  020124  042101  051104
 6514  033554  004456  051105  047522
 6515  033562  020122  042522  027107
 6516  033570     200
 6517
 6518  033571     040  047111  000040  MTA17:  .ASCIZ  ' IN '
 6519
 6520  033576  054105  042520  052103  MTB17:  .ASCIZ  'EXPECTED DATA:'<CRLF>
 6521  033604  042105  042040  052101
 6522  033612  035101  000200
 6523
 6524  033616  054502  042524  004456  MTC17:  .ASCIZ  'BYTE.'<TAB>
 6525  033624     000
 6526
 6527  033625     127  051117  027104  MTA20:  .ASCIZ  'WORD.'<TAB>
 6528  033632  000011
 6529
 6530  033634  054105  042520  052103  MTA21:  .ASCII  'EXPECTED DATA:'<CRLF>
 6531  033642  042105  042040  052101
 6532  033650  035101     200
 6533  033653     110  052111  020123          .ASCIZ  'HITS IN GROUP 0.'<TAB>'/'<TAB>'HITS IN GROUP 1. '<CRLF>
 6534  033660  047111  043440  047522
 6535  033666  050125  030040  004456
 6536  033674  004457  044510  051524
 6537  033702  044440  020116  051107
 6538  033710  052517  020120  027061
 6539  033716  100040     000
 6540
 6541          033571                  MTB21=MTA17
 6542
 6543  033721     200  042524  052123  MTA43:  .ASCII  <CRLF>'TEST ADDRESS.'<TAB>'ERROR ADRS REG.'<TAB>
 6544  033726  040440  042104  042522
 6545  033734  051523  004456  051105
 6546  033742  047522  020122  042101
 6547  033750  051522  051040  043505
 6548  033756  004456
 6549  033760  051105  047522  020122          .ASCIZ  'ERROR REG.'<CRLF>
 6550  033766  042522  027107  000200
 6551
 6552  033774  053600  047522  042524  MTA45:  .ASCIZ  <CRLF>'WROTE. 377'<TAB>'IN BYTE. '
 6553  034002  020056  033463  004467
 6554  034010  047111  041040  052131
 6555  034016  027105  000040
 6556
```

```
 6557  034022  051200  040505  020104  MTB45:  .ASCIZ  <CRLF>'READ DATA. '
 6558  034030  040504  040524  020056
 6559  034036     000
 6560
 6561  034037     011  047111  053440  MTC45:  .ASCIZ  <TAB>'IN WORD. '
 6562  034044  051117  027104  000040
 6563
 6564  034052  053600  047522  042524  MTA50:  .ASCIZ  <CRLF>'WROTE. 000'<TAB>'IN BYTE. '
 6565  034060  020056  030060  004460
 6566  034066  047111  041040  052131
 6567  034074  027105  000040
 6568
 6569  034100  042600  052116  051105  PDMSG1: .ASCII  <CRLF>'ENTERING CACHE ADDRESS MEMORY POWER UP '
 6570  034106  047111  020107  040503
 6571  034114  044103  020105  042101
 6572  034122  051104  051505  020123
 6573  034130  042515  047515  054522
 6574  034136  050040  053517  051105
 6575  034144  052440  020120
 6576  034150  047111  040526  044514          .ASCII  'INVALIDATOR TEST.'<CRLF>
 6577  034156  040504  047524  020122
 6578  034164  042524  052123  100056
 6579  034172  046120  040505  042523          .ASCII  'PLEASE GO THROUGH A POWER DOWN, POWER UP '
 6580  034200  043440  020117  044124
 6581  034206  047522  043525  020110
 6582  034214  020101  047520  042527
 6583  034222  020122  047504  047127
 6584  034230  020054  047520  042527
 6585  034236  020122  050125     040
 6586  034243     123  050505  042525          .ASCIZ  'SEQUENCE.'<CRLF>
 6587  034250  041516  027105  000200
 6588
 6589  034256  041600  041501  042510  PDMSG2: .ASCII  <CRLF>'CACHE ADDRESS MEMORY POWER UP INVALIDATOR'
 6590  034264  040440  042104  042522
 6591  034272  051523  046440  046505
 6592  034300  051117  020131  047520
 6593  034306  042527  020122  050125
 6594  034314  044440  053116  046101
 6595  034322  042111  052101  051117
 6596  034330  052040  051505  020124          .ASCIZ  ' TEST DID NOT FAIL.'<CRLF>
 6597  034336  044504  020104  047516
 6598  034344  020124  040506  046111
 6599  034352  100056     000
 6600
 6601  034355     105  051122  051117  ADRNG:  .ASCII  'ERROR ADDRESS REGISTER NEEDED FOR TEST.'<CRLF>'BUT IT HAS BEEN '
 6602  034362  040440  042104  042522
 6603  034370  051523  051040  043505
 6604  034376  051511  042524  020122
 6605  034404  042516  042105  042105
 6606  034412  043040  051117  052040
 6607  034420  051505  026124  041200
 6608  034426  052125  044440  020124
 6609  034434  040510  020123  042502
 6610  034442  047105     040
 6611  034445     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
 6612  034452  042105  040440  020123
```

```
 6613  034460  040502  020504     000
 6614
 6615  034465     105  051122  051117  ERRNG:  .ASCII  'ERROR REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
 6616  034472  051040  043505  051511
 6617  034500  042524  020122  042516
 6618  034506  042105  042105  043040
 6619  034514  051117  052040  051505
 6620  034522  026124  041200  052125
 6621  034530  044440  020124  040510
 6622  034536  020123  042502  047105
 6623  034544     040
 6624  034545     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
 6625  034552  042105  040440  020123
 6626  034560  040502  020504     000
 6627
 6628  034565     103  047117  051124  CNRNG:  .ASCII  'CONTROL REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
 6629  034572  046117  051040  043505
 6630  034600  051511  042524  020122
 6631  034606  042516  042105  042105
 6632  034614  043040  051117  052040
 6633  034622  051505  026124  041200
 6634  034630  052125  044440  020124
 6635  034636  040510  020123  042502
 6636  034644  047105     040
 6637  034647     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
 6638  034654  042105  040440  020123
 6639  034662  040502  020504     000
 6640  034667     115  044501  052116  MNRNG:  .ASCII  'MAINTENANCE REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
 6641  034674  047105  047101  042503
 6642  034702  051040  043505  051511
 6643  034710  042524  020122  042516
 6644  034716  042105  042105  043040
 6645  034724  051117  052040  051505
 6646  034732  026124  041200  052125
 6647  034740  044440  020124  040510
 6648  034746  020123  042502  047105
 6649  034754     040
 6650  034755     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
 6651  034762  042105  040440  020123
 6652  034770  040502  020504     000
 6653
 6654  034775     110  052111  046457  HMRNG:  .ASCII  'HIT/MISS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
 6655  035002  051511  020123  042522
 6656  035010  044507  052123  051105
 6657  035016  047040  042505  042504
 6658  035024  020104  047506  020122
 6659  035032  042524  052123  100054
 6660  035040  052502  020124  052111
 6661  035046  044040  051501  041040
 6662  035054  042505  020116
 6663  035060  046106  043501  042507          .ASCIZ  'FLAGGED AS BAD!'
 6664  035066  020104  051501  041040
 6665  035074  042101  000041
 6666
 6667  035100  040600  042104  042522  MTA77:  .ASCIZ  <CRLF>'ADDRESS:  '
 6668  035106  051523  020072  000040
```

```
 6669
 6670  035114  051440  047510  046125  MTB77:  .ASCIZ  ' SHOULD HAVE BEEN A HIT IN GROUP '
 6671  035122  020104  040510  042526
 6672  035130  041040  042505  020116
 6673  035136  020101  044510  020124
 6674  035144  047111  043440  047522
 6675  035152  050125  000040
 6676
 6677  035156  043101  042524  020122  MTC77:  .ASCIZ  'AFTER REFERENCING'<CRLF>'ADDRESS:  '
 6678  035164  042522  042506  042522
 6679  035172  041516  047111  100107
 6680  035200  042101  051104  051505
 6681  035206  035123  020040     000
 6682
 6683  035213     040  044127  046111  MTD77:  .ASCIZ  ' WHILE FORCING SELECTION OF GROUP '
 6684  035220  020105  047506  041522
 6685  035226  047111  020107  042523
 6686  035234  042514  052103  047511
 6687  035242  020116  043117  043440
 6688  035250  047522  050125  000040
 6689
 6690  035256  040600  051122  051117  MTA101: .ASCII  <CRLF>'ARROR ADRS REG.'<TAB>'ERROR REG.'<TAB>
 6691  035264  040440  051104  020123
 6692  035272  042522  027107  042411
 6693  035300  051122  051117  051040
 6694  035306  043505  004456
 6695  035312  054105  042520  052103          .ASCIZ  'EXPECTED ERR.'<TAB>'PATTERN PUT IN MAINT REG.'<CRLF>
 6696  035320  042105  042440  051122
 6697  035326  004456  040520  052124
 6698  035334  051105  020116  052520
 6699  035342  020124  047111  046440
 6700  035350  044501  052116  051040
 6701  035356  043505  100056     000
 6702
 6703  035363     200  043101  042524  MTA120: .ASCIZ  <CRLF>'AFTER 2ND CYCLE READ  '
 6704  035370  020122  047062  020104
 6705  035376  054503  046103  020105
 6706  035404  042522  042101  020040
 6707  035412     000
 6708
 6709  035413     200  043101  042524  MTB120: .ASCIZ  <CRLF>'AFTER 4TH CYCLE READ  '
 6710  035420  020122  052064  020110
 6711  035426  054503  046103  020105
 6712  035434  042522  042101  020040
 6713  035442     000
 6714
 6715  035443     200  043101  042524  MTC120: .ASCIZ  <CRLF>'AFTER 6TH CYCLE READ  '
 6716  035450  020122  052066  020110
 6717  035456  054503  046103  020105
 6718  035464  042522  042101  020040
 6719  035472     000
 6720  035473     200  043101  042524  MTD120: .ASCIZ  <CRLF>'AFTER 8TH CYCLE READ  '
 6721  035500  020122  052070  020110
 6722  035506  054503  046103  020105
 6723  035514  042522  042101  020040
 6724  035522     000
```

```
6725
6726  035523     200  043101  042524  MTE120: .ASCIZ  <CRLF>'AFTER 10TH CYCLE READ '
6727  035530  020122  030061  044124
6728  035536  041440  041531  042514
6729  035544  051040  040505  020104
6730  035552     000
6731
6732  035553     200  043101  042524  MTF120: .ASCIZ  <CRLF>'AFTER 12TH CYCLE READ '
6733  035560  020122  031061  044124
6734  035566  041440  041531  042514
6735  035574  051040  040505  020104
6736  035602     000
6737
6738  035603     106  047522  020115  MTG120: .ASCIZ  'FROM THE HIT/MISS REG. EXPECTED '
6739  035610  044124  020105  044510
6740  035616  027524  044515  051523
6741  035624  051040  043505  020056
6742  035632  054105  042520  052103
6743  035640  042105  000040
6744
6745  035644  052200  042510  050040  MTA124: .ASCII  <CRLF>'THE PATTERN BEING USED IN THE MAINTENANCE ' ;
6746  035652  052101  042524  047122
6747  035660  041040  044505  043516
6748  035666  052440  042523  020104
6749  035674  047111  052040  042510
6750  035702  046440  044501  052116
6751  035710  047105  047101  042503
6752  035716     040
6753  035717     122  043505  051511          .ASCIZ  'REGISTER WAS: '
6754  035724  042524  020122  040527
6755  035732  035123  000040
6756
6757  035736  051200  043105  051105  MTA126: .ASCIZ  <CRLF>'REFERENCED ADDRESS:'<TAB>
6758  035744  047105  042503  020104
6759  035752  042101  051104  051505
6760  035760  035123  000011
6761
6762  035764  040600  051122  051117  MTB126: .ASCIZ  <CRLF>'ARROR ADDRESS REGISTER:'<TAB>
6763  035772  040440  042104  042522
6764  036000  051523  051040  043505
6765  036006  051511  042524  035122
6766  036014  000011
6767
6768  036016  050200  052101  042524  MTA131: .ASCIZ  <CRLF>'PATTERN BEING USED IN THE MAINTENANCE REGISTER:'<TAB>
6769  036024  047122  041040  044505
6770  036032  043516  052440  042523
6771  036040  020104  047111  052040
6772  036046  042510  046440  044501
6773  036054  052116  047105  047101
6774  036062  042503  051040  043505
6775  036070  051511  042524  035122
6776  036076  000011
6777
6778  036100  042600  050130  041505  MTB131: .ASCIZ  <CRLF>'EXPECTED ERROR REGISTER:'<TAB>
6779  036106  042524  020104  051105
6780  036114  047522  020122  042522
```

```
 6781  036122  044507  052123  051105
 6782  036130  004472     000
 6783
 6784  036133     200  047507  020124  MTC131: .ASCIZ  <CRLF>'GOT ERROR REGISTER:'<TAB>
 6785  036140  051105  047522  020122
 6786  036146  042522  044507  052123
 6787  036154  051105  004472     000
 6788
 6789  036161     200  051105  047522  MTA134: .ASCIZ  <CRLF>'ERROR ADR REG.'<TAB>'ERROR REG.'<CRLF>
 6790  036166  020122  042101  020122
 6791  036174  042522  027107  042411
 6792  036202  051122  051117  051040
 6793  036210  043505  100056     000
 6794
 6795  036215     200  054105  042520  MTA135: .ASCIZ  <CRLF>'EXPECTED ERROR REG.:  '
 6796  036222  052103  042105  042440
 6797  036230  051122  051117  051040
 6798  036236  043505  035056  020040
 6799  036244     000
 6800
 6801  036245     107  052117  042440  MTB135: .ASCIZ  'GOT ERROR REG.:  '
 6802  036252  051122  051117  051040
 6803  036260  043505  035056  020040
 6804  036266     000
 6805
 6806  036267     200  054105  042520  MTC135: .ASCIZ  <CRLF>'EXPECTED ERROR ADR REG.:  '
 6807  036274  052103  042105  042440
 6808  036302  051122  051117  040440
 6809  036310  051104  051040  043505
 6810  036316  035056  020040     000
 6811
 6812  036323     107  052117  042440  MTD135: .ASCIZ  'GOT ERROR ADR REG.:  '
 6813  036330  051122  051117  040440
 6814  036336  051104  051040  043505
 6815  036344  035056  020040     000
 6816  036351     200  050103  020125  MSG1:   .ASCIZ<CRLF>    ''CPU UNDER TEST FOUND TO BE A ''
 6817  036356  047125  042504  020122
 6818  036364  042524  052123  043040
 6819  036372  052517  042116  052040
 6820  036400  020117  042502  040440
 6821  036406  000040
 6822  036410  041113  030461  042455  MSG2:   .ASCIZ  '%B11-EM'<CRLF>
 6823  036416  100115     000
 6824  036421     113  030502  026461  MSG3:   .ASCIZ  '%B11-B/C'<CRLF>
 6825  036426  027502  100103     000
 6826  036433     113  030502  026461  MSG4:   .ASCIZ  '%B11-CM                  ''<CRLF>
 6827  036440  046503  020040  020040
 6828  036446  020040  020040  020040
 6829  036454  020040  020040  020040
 6830  036462  000200
 6831  036464  041113  030461  042455  MSG5:   .ASCIZ  '%B11-E''<CRLF>
 6832  036472  000200
 6833
 6834                                  ;THESE ARE THE ERROR MESSAGES:
 6835
 6836  036474  020101  042522  042506  EM1:    .ASCIZ  'A REFERENCE WHICH SHOULD HAVE BEEN A HIT WAS A MISS.'
```

```
 6837  036502  042522  041516  020105
 6838  036510  044127  041511  020110
 6839  036516  044123  052517  042114
 6840  036524  044040  053101  020105
 6841  036532  042502  047105  040440
 6842  036540  044040  052111  053440
 6843  036546  051501  040440  046440
 6844  036554  051511  027123     000
 6845
 6846
 6847  036561     200  047125  054105  EM14:   .ASCIZ  <CRLF>'UNEXPECTED PARITY ERROR TRAP.'
 6848  036566  042520  052103  042105
 6849  036574  050040  051101  052111
 6850  036602  020131  051105  047522
 6851  036610  020122  051124  050101
 6852  036616  000056
 6853
 6854  036620  025052  052052  051505  EM15:   .ASCIZ  '***TEST ABORTED! GOING TO NEXT TEST.***'
 6855  036626  020124  041101  051117
 6856  036634  042524  020504  043440
 6857  036642  044517  043516  052040
 6858  036650  020117  042516  052130
 6859  036656  052040  051505  027124
 6860  036664  025052  000052
 6861  036670  040503  044103  020105  EM55:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6862  036676  042522  044507  052123
 6863  036704  051105  051040  051505
 6864  036712  047520  051516  020105
 6865  036720  042524  052123  043040
 6866  036726  044501  042514  027104
 6867  036734     200
 6868  036735     101  051040  043105          .ASCII  'A REFERENCE TO THE LOW ORDER ERROR ADDRESS REGISTER '
 6869  036742  051105  047105  042503
 6870  036750  052040  020117  044124
 6871  036756  020105  047514  020127
 6872  036764  051117  042504  020122
 6873  036772  051105  047522  020122
 6874  037000  042101  051104  051505
 6875  037006  020123  042522  044507
 6876  037014  052123  051105     040
 6877  037021     124  046511  042105          .ASCIZ  'TIMED OUT.'
 6878  037026  047440  052125  000056
 6879
 6880  037034  040503  044103  020105  EM56:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6881  037042  042522  044507  052123
 6882  037050  051105  051040  051505
 6883  037056  047520  051516  020105
 6884  037064  042524  052123  043040
 6885  037072  044501  042514  027104
 6886  037100     200
 6887  037101     101  051040  043105          .ASCII  'A REFERENCE TO THE HIGH ORDER ERROR ADDRESS REGISTER '
 6888  037106  051105  047105  042503
 6889  037114  052040  020117  044124
 6890  037122  020105  044510  044107
 6891  037130  047440  042122  051105
 6892  037136  042440  051122  051117
```

```
 6893  037144  040440  042104  042522
 6894  037152  051523  051040  043505
 6895  037160  051511  042524  020122
 6896  037166  044524  042515  020104          .ASCIZ  'TIMED OUT.'
 6897  037174  052517  027124     000
 6898
 6899  037201     103  041501  042510  EM57:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6900  037206  051040  043505  051511
 6901  037214  042524  020122  042522
 6902  037222  050123  047117  042523
 6903  037230  052040  051505  020124
 6904  037236  040506  046111  042105
 6905  037244  100056
 6906  037246  020101  042522  042506          .ASCIZ  'A REFERENCE TO THE ERROR REGISTER TIMED OUT.'
 6907  037254  042522  041516  020105
 6908  037262  047524  052040  042510
 6909  037270  042440  051122  051117
 6910  037276  051040  043505  051511
 6911  037304  042524  020122  044524
 6912  037312  042515  020104  052517
 6913  037320  027124     000
 6914
 6915  037323     103  041501  042510  EM60:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6916  037330  051040  043505  051511
 6917  037336  042524  020122  042522
 6918  037344  050123  047117  042523
 6919  037352  052040  051505  020124
 6920  037360  040506  046111  042105
 6921  037366  100056
 6922  037370  020101  042522  042506          .ASCIZ  'A REFERENCE TO THE CONTROL REGISTER TIMED OUT.'
 6923  037376  042522  041516  020105
 6924  037404  047524  052040  042510
 6925  037412  041440  047117  051124
 6926  037420  046117  051040  043505
 6927  037426  051511  042524  020122
 6928  037434  044524  042515  020104
 6929  037442  052517  027124     000
 6930
 6931  037447     103  041501  042510  EM61:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6932  037454  051040  043505  051511
 6933  037462  042524  020122  042522
 6934  037470  050123  047117  042523
 6935  037476  052040  051505  020124
 6936  037504  040506  046111  042105
 6937  037512  100056
 6938  037514  020101  042522  042506          .ASCIZ  'A REFERENCE TO THE MAINTENANCE REGISTER TIMED OUT.'
 6939  037522  042522  041516  020105
 6940  037530  047524  052040  042510
 6941  037536  046440  044501  052116
 6942  037544  047105  047101  042503
 6943  037552  051040  043505  051511
 6944  037560  042524  020122  044524
 6945  037566  042515  020104  052517
 6946  037574  027124     000
 6947
 6948  037577     103  041501  042510  EM62:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
```

```
6949  037604  051040  043505  051511
6950  037612  042524  020122  042522
6951  037620  050123  047117  042523
6952  037626  052040  051505  020124
6953  037634  040506  046111  042105
6954  037642  100056
6955  037644  020101  042522  042506          .ASCIZ  'A REFERENCE TO THE HIT/MISS REGISTER TIMED OUT.'<CRLF>
6956  037652  042522  041516  020105
6957  037660  047524  052040  042510
6958  037666  044040  052111  046457
6959  037674  051511  020123  042522
6960  037702  044507  052123  051105
6961  037710  052040  046511  042105
6962  037716  047440  052125  100056
6963  037724     000
6964
6965  037725     103  041501  042510  EM63:   .ASCII  'CACHE REGISTER DATA PATHS, READ ZEROES, TEST FAILED.'
6966  037732  051040  043505  051511
6967  037740  042524  020122  040504
6968  037746  040524  050040  052101
6969  037754  051510  020054  042522
6970  037762  042101  055040  051105
6971  037770  042517  026123  052040
6972  037776  051505  020124  040506
6973  040004  046111  042105     056
6974  040011     200  051127  052117          .ASCII  <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA '
6975  040016  020105  042532  047522
6976  040024  051505  041040  052125
6977  040032  051040  040505  020104
6978  040040  040502  045503  047040
6979  040046  047117  055055  051105
6980  040054  020117  040504  040524
6981  040062     040
6982  040063     106  047522  020115          .ASCIZ  'FROM BOTH'<CRLF>'THE CONTROL AND MAINTENANCE REGISTERS.'
6983  040070  047502  044124  052200
6984  040076  042510  041440  047117
6985  040104  051124  046117  040440
6986  040112  042116  046440  044501
6987  040120  052116  047105  047101
6988  040126  042503  051040  043505
6989  040134  051511  042524  051522
6990  040142  000056
6991
6992  040144  040503  044103  020105  EM64:   .ASCII  'CACHE REGISTER DATA PATH, READ ZEROES, TEST FAILED.'
6993  040152  042522  044507  052123
6994  040160  051105  042040  052101
6995  040166  020101  040520  044124
6996  040174  020054  042522  042101
6997  040202  055040  051105  042517
6998  040210  026123  052040  051505
6999  040216  020124  040506  046111
7000  040224  042105     056
7001  040227     200  051127  052117          .ASCII  <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA FROM '
7002  040234  020105  042532  047522
7003  040242  051505  041040  052125
7004  040250  051040  040505  020104
```

```
 7005  040256  040502  045503  047040
 7006  040264  047117  055055  051105
 7007  040272  020117  040504  040524
 7008  040300  043040  047522  020115
 7009  040306  052200  042510  041440          .ASCIZ  <CRLF>'THE CACHE CONTROL REGISTER.'
 7010  040314  041501  042510  041440
 7011  040322  047117  051124  046117
 7012  040330  051040  043505  051511
 7013  040336  042524  027122     000
 7014
 7015  040343     103  041501  042510  EM65:   .ASCII  'CACHE REGISTER DATA PATHS, READ ONES, REST FAILED.'<CRLF>
 7016  040350  051040  043505  051511
 7017  040356  042524  020122  040504
 7018  040364  040524  050040  052101
 7019  040372  051510  020054  042522
 7020  040400  042101  047440  042516
 7021  040406  026123  051040  051505
 7022  040414  020124  040506  046111
 7023  040422  042105  100056
 7024  040426  040506  046111  042105          .ASCII  'FAILED TO READ CORRECT DATA FROM THE ADDRESS REGISTER'
 7025  040434  052040  020117  042522
 7026  040442  042101  041440  051117
 7027  040450  042522  052103  042040
 7028  040456  052101  020101  051106
 7029  040464  046517  052040  042510
 7030  040472  040440  042104  042522
 7031  040500  051523  051040  043505
 7032  040506  051511  042524     122
 7033  040513     040  047111  052040          .ASCII  ' IN THE CLEAR STATE.'<CRLF>'THE LOW ORDER ADDRESS '
 7034  040520  042510  041440  042514
 7035  040526  051101  051440  040524
 7036  040534  042524  100056  044124
 7037  040542  020105  047514  020127
 7038  040550  051117  042504  020122
 7039  040556  042101  051104  051505
 7040  040564  020123
 7041  040566  044123  052517  042114          .ASCII  'SHOULD HAVE BEEN SET TO: 177740'<CRLF>
 7042  040574  044040  053101  020105
 7043  040602  042502  047105  051440
 7044  040610  052105  052040  035117
 7045  040616  030440  033467  032067
 7046  040624  100060
 7047  040626  044124  020105  044510          .ASCII  'THE HIGH ORDER ADDRESS REGISTER SHOULD HAVE BEEN '
 7048  040634  044107  047440  042122
 7049  040642  051105  040440  042104
 7050  040650  042522  051523  051040
 7051  040656  043505  051511  042524
 7052  040664  020122  044123  052517
 7053  040672  042114  044040  053101
 7054  040700  020105  042502  047105
 7055  040706     040
 7056  040707     123  052105  052040          .ASCIZ  'SET TO: 000003'
 7057  040714  035117  030040  030060
 7058  040722  030060  000063
 7059
 7060  040726  040503  044103  020105  EM66:   .ASCIZ  'CACHE CONTROL REGISTER COUNT PATTERN TEST FAILED.'
```

```
 7061  040734  047503  052116  047522
 7062  040742  020114  042522  044507
 7063  040750  052123  051105  041440
 7064  040756  052517  052116  050040
 7065  040764  052101  042524  047122
 7066  040772  052040  051505  020124
 7067  041000  040506  046111  042105
 7068  041006  000056
 7069
 7070  041010  040503  044103  020105  EM67:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 7071  041016  044510  027524  044515
 7072  041024  051523  040440  042116
 7073  041032  041440  047117  051124
 7074  041040  046117  051040  043505
 7075  041046  051511  042524  020122
 7076  041054  042524  052123  043040
 7077  041062  044501  042514  027104
 7078  041070  053600  052111  020110          .ASCII  <CRLF>'WITH THE CONTROL REGISTER CLEAR, THE HIT/MISS '
 7079  041076  044124  020105  047503
 7080  041104  052116  047522  020114
 7081  041112  042522  044507  052123
 7082  041120  051105  041440  042514
 7083  041126  051101  020054  044124
 7084  041134  020105  044510  027524
 7085  041142  044515  051523     040
 7086  041147     122  043505  051511          .ASCIZ  'REGISTER SHOULD'<CRLF>'HAVE SHOWN SIX HITS (000077).'
 7087  041154  042524  020122  044123
 7088  041162  052517  042114  044200
 7089  041170  053101  020105  044123
 7090  041176  053517  020116  044523
 7091  041204  020130  044510  051524
 7092  041212  024040  030060  030060
 7093  041220  033467  027051     000
 7094
 7095  041225     103  041501  042510  EM70:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 7096  041232  044040  052111  046457
 7097  041240  051511  020123  047101
 7098  041246  020104  047503  052116
 7099  041254  047522  020114  042522
 7100  041262  044507  052123  051105
 7101  041270  052040  051505  020124
 7102  041276  040506  046111  042105
 7103  041304     056
 7104  041305     200  044127  046111          .ASCII  <CRLF>'WHILE FORCING SELECTION OF GROUP 1 AND FORCING '
 7105  041312  020105  047506  041522
 7106  041320  047111  020107  042523
 7107  041326  042514  052103  047511
 7108  041334  020116  043117  043440
 7109  041342  047522  050125  030440
 7110  041350  040440  042116  043040
 7111  041356  051117  044503  043516
 7112  041364     040
 7113  041365     115  051511  042523          .ASCII  'MISSES TO GROUP 0,'<CRLF>'THE HIT/MISS REGISTER '
 7114  041372  020123  047524  043440
 7115  041400  047522  050125  030040
 7116  041406  100054  044124  020105
```

```
 7117  041414  044510  027524  044515
 7118  041422  051523  051040  043505
 7119  041430  051511  042524  020122
 7120  041436  044123  052517  042114          .ASCIZ  'SHOULD HAVE SHOWN SIX HITS (000077).'
 7121  041444  044040  053101  020105
 7122  041452  044123  053517  020116
 7123  041460  044523  020130  044510
 7124  041466  051524  024040  030060
 7125  041474  030060  033467  027051
 7126  041502     000
 7127
 7128  041503     103  041501  042510  EM71:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 7129  041510  044040  052111  046457
 7130  041516  051511  020123  047101
 7131  041524  020104  047503  052116
 7132  041532  047522  020114  042522
 7133  041540  044507  052123  051105
 7134  041546  052040  051505  020124
 7135  041554  040506  046111  042105
 7136  041562     056
 7137  041563     200  044127  046111          .ASCII  <CRLF>'WHILE FORCING SELECTION OF GROUP 0 AND FORCING '
 7138  041570  020105  047506  041522
 7139  041576  047111  020107  042523
 7140  041604  042514  052103  047511
 7141  041612  020116  043117  043440
 7142  041620  047522  050125  030040
 7143  041626  040440  042116  043040
 7144  041634  051117  044503  043516
 7145  041642     040
 7146  041643     115  051511  042523          .ASCII  'MISSES TO GROUP 1,'<CRLF>'THE HIT/MISS REGISTER '
 7147  041650  020123  047524  043440
 7148  041656  047522  050125  030440
 7149  041664  100054  044124  020105
 7150  041672  044510  027524  044515
 7151  041700  051523  051040  043505
 7152  041706  051511  042524  020122
 7153  041714  044123  052517  042114          .ASCIZ  'SHOULD HAVE SHOWN SIX HITS (000077).'
 7154  041722  044040  053101  020105
 7155  041730  044123  053517  020116
 7156  041736  044523  020130  044510
 7157  041744  051524  024040  030060
 7158  041752  030060  033467  027051
 7159  041760     000
 7160
 7161  041761     103  041501  042510  EM72:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 7162  041766  044040  052111  046457
 7163  041774  051511  020123  047101
 7164  042002  020104  047503  052116
 7165  042010  047522  020114  042522
 7166  042016  044507  052123  051105
 7167  042024  052040  051505  020124
 7168  042032  040506  046111  042105
 7169  042040     056
 7170  042041     127  044510  042514          .ASCII  'WHILE FORCING MISSES TO BOTH GROUPS. THE HIT/MISS '
 7171  042046  043040  051117  044503
 7172  042054  043516  046440  051511
```

```
 7173  042062  042523  020123  047524
 7174  042070  041040  052117  020110
 7175  042076  051107  052517  051520
 7176  042104  020054  044124  020105
 7177  042112  044510  027524  044515
 7178  042120  051523     040
 7179  042123     122  043505  051511          .ASCIZ  'REGISTER'<CRLF>'SHOULD HAVE SHOWN SIX MISSES (000000).'
 7180  042130  042524  100122  044123
 7181  042136  052517  042114  044040
 7182  042144  053101  020105  044123
 7183  042152  053517  020116  044523
 7184  042160  020130  044515  051523
 7185  042166  051505  024040  030060
 7186  042174  030060  030060  027051
 7187  042202     000
 7188
 7189  042203     103  041501  042510  EM73:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 7190  042210  044040  052111  046457
 7191  042216  051511  020123  047101
 7192  042224  020104  047503  052116
 7193  042232  047522  020114  042522
 7194  042240  044507  052123  051105
 7195  042246  052040  051505  020124
 7196  042254  040506  046111  042105
 7197  042262     056
 7198  042263     200  044127  046111          .ASCII  <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '
 7199  042270  020105  047506  041522
 7200  042276  047111  020107  044515
 7201  042304  051523  051505  052040
 7202  042312  020117  047502  044124
 7203  042320  043440  047522  050125
 7204  042326  020123  047101  020104
 7205  042334  047506  041522  047111
 7206  042342  020107
 7207  042344  042523  042514  052103          .ASCII  'SELECTION OF GROUP 1,'<CRLF>'THE HIT/MISS REGISTER '
 7208  042352  047511  020116  043117
 7209  042360  043440  047522  050125
 7210  042366  030440  100054  044124
 7211  042374  020105  044510  027524
 7212  042402  044515  051523  051040
 7213  042410  043505  051511  042524
 7214  042416  020122
 7215  042420  044123  052517  042114          .ASCIZ  'SHOULD HAVE SHOWN SIX MISSES (000000).'
 7216  042426  044040  053101  020105
 7217  042434  044123  053517  020116
 7218  042442  044523  020130  044515
 7219  042450  051523  051505  024040
 7220  042456  030060  030060  030060
 7221  042464  027051     000
 7222
 7223  042467     103  041501  042510  EM74:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 7224  042474  044040  052111  046457
 7225  042502  051511  020123  047101
 7226  042510  020104  047503  052116
 7227  042516  047522  020114  042522
 7228  042524  044507  052123  051105
```

```
 7229  042532  052040  051505  020124
 7230  042540  040506  046111  042105
 7231  042546     056
 7232  042547     200  044127  046111          .ASCII  <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '
 7233  042554  020105  047506  041522
 7234  042562  047111  020107  044515
 7235  042570  051523  051505  052040
 7236  042576  020117  047502  044124
 7237  042604  043440  047522  050125
 7238  042612  020123  047101  020104
 7239  042620  047506  041522  047111
 7240  042626  020107
 7241  042630  042523  042514  052103          .ASCII  'SELECTION OF GROUP 0,'<CRLF>'THE HIT/MISS REGISTER '
 7242  042636  047511  020116  043117
 7243  042644  043440  047522  050125
 7244  042652  030040  100054  044124
 7245  042660  020105  044510  027524
 7246  042666  044515  051523  051040
 7247  042674  043505  051511  042524
 7248  042702  020122
 7249  042704  044123  052517  042114          .ASCIZ  'SHOULD HAVE SHOWN SIX MISSES (000000).'
 7250  042712  044040  053101  020105
 7251  042720  044123  053517  020116
 7252  042726  044523  020130  044515
 7253  042734  051523  051505  024040
 7254  042742  030060  030060  030060
 7255  042750  027051     000
 7256
 7257  042753     103  047117  051124  EM75:   .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>'FAILED TO GET '
 7258  042760  046117  051040  043505
 7259  042766  051511  042524  020122
 7260  042774  042524  052123  043040
 7261  043002  044501  042514  027104
 7262  043010  043200  044501  042514
 7263  043016  020104  047524  043440
 7264  043024  052105     040
 7265  043027     101  044040  052111          .ASCIZ  'A HIT ON A REFERENCE WHICH SHOULD HAVE BEEN A HIT.'
 7266  043034  047440  020116  020101
 7267  043042  042522  042506  042522
 7268  043050  041516  020105  044127
 7269  043056  041511  020110  044123
 7270  043064  052517  042114  044040
 7271  043072  053101  020105  042502
 7272  043100  047105  040440  044040
 7273  043106  052111  000056
 7274
 7275          042753                  EM76=EM75
 7276
 7277  043112  047503  052116  047522  EM77:   .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>'THE WRONG '
 7278  043120  020114  042522  044507
 7279  043126  052123  051105  052040
 7280  043134  051505  020124  040506
 7281  043142  046111  042105  100056
 7282  043150  044124  020105  051127
 7283  043156  047117  020107
 7284  043162  051107  052517  020120          .ASCIZ  'GROUP WAS WRITTEN WHILE FORCING SELECTION OF A GROUP.'
```

```
 7285  043170  040527  020123  051127
 7286  043176  052111  042524  020116
 7287  043204  044127  046111  020105
 7288  043212  047506  041522  047111
 7289  043220  020107  042523  042514
 7290  043226  052103  047511  020116
 7291  043234  043117  040440  043440
 7292  043242  047522  050125  000056
 7293
 7294  043250  047503  052116  047522  EM117:  .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>
 7295  043256  020114  042522  044507
 7296  043264  052123  051105  052040
 7297  043272  051505  020124  040506
 7298  043300  046111  042105  100056
 7299  043306  047507  020124  020101          .ASCIZ  'GOT A HIT IN THE GROUP TO WHICH MISSES ARE BEING FORCED.'
 7300  043314  044510  020124  047111
 7301  043322  052040  042510  043440
 7302  043330  047522  050125  052040
 7303  043336  020117  044127  041511
 7304  043344  020110  044515  051523
 7305  043352  051505  040440  042522
 7306  043360  041040  044505  043516
 7307  043366  043040  051117  042503
 7308  043374  027104     000
 7309
 7310  043377     110  052111  046457  EM120:  .ASCII  'HIT/MISS REGISTER PATTERNS TEST FAILED.'
 7311  043404  051511  020123  042522
 7312  043412  044507  052123  051105
 7313  043420  050040  052101  042524
 7314  043426  047122  020123  042524
 7315  043434  052123  043040  044501
 7316  043442  042514  027104
 7317  043446  051200  040505  020104          .ASCII  <CRLF>'READ WRONG DATA FROM THE HIT/MISS REGISTER'<CRLF>
 7318  043454  051127  047117  020107
 7319  043462  040504  040524  043040
 7320  043470  047522  020115  044124
 7321  043476  020105  044510  027524
 7322  043504  044515  051523  051040
 7323  043512  043505  051511  042524
 7324  043520  100122
 7325  043522  044127  046111  020105          .ASCIZ  'WHILE FLOATING A PATTERN OF HITS AND MISSES THROUGH IT.'
 7326  043530  046106  040517  044524
 7327  043536  043516  040440  050040
 7328  043544  052101  042524  047122
 7329  043552  047440  020106  044510
 7330  043560  051524  040440  042116
 7331  043566  046440  051511  042523
 7332  043574  020123  044124  047522
 7333  043602  043525  020110  052111
 7334  043610  000056
 7335
 7336  043612  040503  044103  020105  EM121:  .ASCII  /CACHE CONTROL SIGNAL, THE 'RANDOM' SIGNAL, TEST FAILED./
 7337  043620  047503  052116  047522
 7338  043626  020114  044523  047107
 7339  043634  046101  020054  044124
 7340  043642  020105  051047  047101
```

```
7341  043650  047504  023515  051440
7342  043656  043511  040516  026114
7343  043664  052040  051505  020124
7344  043672  040506  046111  042105
7345  043700     056
7346  043701     200  040506  046111          .ASCII  <CRLF>'FAILED TO GET BOTH HITS AT THE TWO TEST ADDRESSES '
7347  043706  042105  052040  020117
7348  043714  042507  020124  047502
7349  043722  044124  044040  052111
7350  043730  020123  052101  052040
7351  043736  042510  052040  047527
7352  043744  052040  051505  020124
7353  043752  042101  051104  051505
7354  043760  042523  020123
7355  043764  044127  041511  020110          .ASCIZ  'WHICH WERE REFERENCED.'
7356  043772  042527  042522  051040
7357  044000  043105  051105  047105
7358  044006  042503  027104     000
7359
7360  044013     115  044501  052116  EM122:  .ASCII  'MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'
7361  044020  047105  047101  042503
7362  044026  051040  043505  051511
7363  044034  042524  020122  047503
7364  044042  047125  020124  040520
7365  044050  052124  051105  020116
7366  044056  042524  052123  043040
7367  044064  044501  042514  027104
7368  044072  052200  042510  046440          .ASCII  <CRLF>'THE MAINTENANCE REGISTER WILL NOT CLEAR.'
7369  044100  044501  052116  047105
7370  044106  047101  042503  051040
7371  044114  043505  051511  042524
7372  044122  020122  044527  046114
7373  044130  047040  052117  041440
7374  044136  042514  051101     056
7375
7376  044143     103  041501  042510  EM123:  .ASCII  'CACHE MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'
7377  044150  046440  044501  052116
7378  044156  047105  047101  042503
7379  044164  051040  043505  051511
7380  044172  042524  020122  047503
7381  044200  047125  020124  040520
7382  044206  052124  051105  020116
7383  044214  042524  052123  043040
7384  044222  044501  042514  027104
7385  044230  040600  052106  051105          .ASCII  <CRLF>'AFTER WRITING A PATTERN IN THIS REGISTER '
7386  044236  053440  044522  044524
7387  044244  043516  040440  050040
7388  044252  052101  042524  047122
7389  044260  044440  020116  044124
7390  044266  051511  051040  043505
7391  044274  051511  042524  020122
7392  044302  040506  046111  042105          .ASCIZ  'FAILED TO READ THAT PATTERN BACK.'
7393  044310  052040  020117  042522
7394  044316  042101  052040  040510
7395  044324  020124  040520  052124
7396  044332  051105  020116  040502
```

```
 7397  044340  045503  000056
 7398
 7399  044344  047101  052440  042516  EM124:  .ASCII  'AN UNEXPECTED ERROR OCCURRED WHILE RUNNING THE '
 7400  044352  050130  041505  042524
 7401  044360  020104  051105  047522
 7402  044366  020122  041517  052503
 7403  044374  051122  042105  053440
 7404  044402  044510  042514  051040
 7405  044410  047125  044516  043516
 7406  044416  052040  042510     040
 7407  044423     115  044501  052116          .ASCII  'MAINTENANCE REGISTER'<CRLF>'COUNT PATTERN '
 7408  044430  047105  047101  042503
 7409  044436  051040  043505  051511
 7410  044444  042524  100122  047503
 7411  044452  047125  020124  040520
 7412  044460  052124  051105  020116
 7413  044466  042524  052123  020056          .ASCIZ  'TEST. NOTE MISSES WERE BEING FORCED TO BOTH GROUPS.'
 7414  044474  047516  042524  046440
 7415  044502  051511  042523  020123
 7416  044510  042527  042522  041040
 7417  044516  044505  043516  043040
 7418  044524  051117  042503  020104
 7419  044532  047524  041040  052117
 7420  044540  020110  051107  052517
 7421  044546  051520  000056
 7422
 7423  044552  040515  047111  042524  EM127:  .ASCII  'MAINTENANCE REGISTER TEST FAILED.'<CRLF>
 7424  044560  040516  041516  020105
 7425  044566  042522  044507  052123
 7426  044574  051105  052040  051505
 7427  044602  020124  040506  046111
 7428  044610  042105  100056
 7429  044614  047516  052040  040522          .ASCII  'NO TRAP OR ABORT OCCURRED WHEN THE PATTERN WAS PUT '
 7430  044622  020120  051117  040440
 7431  044630  047502  052122  047440
 7432  044636  041503  051125  042522
 7433  044644  020104  044127  047105
 7434  044652  052040  042510  050040
 7435  044660  052101  042524  047122
 7436  044666  053440  051501  050040
 7437  044674  052125     040
 7438  044677     111  020116  044124          .ASCIZ  'IN THE MAINTENANCE REGISTER.'
 7439  044704  020105  040515  047111
 7440  044712  042524  040516  041516
 7441  044720  020105  042522  044507
 7442  044726  052123  051105  000056
 7443
 7444  044734  051105  047522  020122  EM130:  .ASCIZ  'ERROR REGISTER WILL NOT UNLOCK, OR CLEAR.'
 7445  044742  042522  044507  052123
 7446  044750  051105  053440  046111
 7447  044756  020114  047516  020124
 7448  044764  047125  047514  045503
 7449  044772  020054  051117  041440
 7450  045000  042514  051101  000056
 7451
 7452  045006  051105  047522  020122  EM131:  .ASCII  'ERROR REGISTER AND MAINTENANCE REGISTER TEST FAILED.'
```

```
 7453  045014  042522  044507  052123
 7454  045022  051105  040440  042116
 7455  045030  046440  044501  052116
 7456  045036  047105  047101  042503
 7457  045044  051040  043505  051511
 7458  045052  042524  020122  042524
 7459  045060  052123  043040  044501
 7460  045066  042514  027104
 7461  045072  042600  051122  051117          .ASCII  <CRLF>'ERROR REGISTER IS INCORRECTLY SET'
 7462  045100  051040  043505  051511
 7463  045106  042524  020122  051511
 7464  045114  044440  041516  051117
 7465  045122  042522  052103  054514
 7466  045130  051440  052105
 7467  045134  043200  051117  052040          .ASCIZ  <CRLF>'FOR THE ERROR THAT WAS FORCED USING THE MAINTENANCE REGISTER.'
 7468  045142  042510  042440  051122
 7469  045150  051117  052040  040510
 7470  045156  020124  040527  020123
 7471  045164  047506  041522  042105
 7472  045172  052440  044523  043516
 7473  045200  052040  042510  046440
 7474  045206  044501  052116  047105
 7475  045214  047101  042503  051040
 7476  045222  043505  051511  042524
 7477  045230  027122     000
 7478
 7479  045233                  EM140:
 7480  045233     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 7481  045240  042515  047515  054522
 7482  045246  042040  052101  020101
 7483  045254  040520  044522  054524
 7484  045262  041440  042510  045503
 7485  045270  051105  020123  042524
 7486  045276  052123  043040  044501
 7487  045304  042514  027104
 7488  045310  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 7489  045316  020105  047524  043040
 7490  045324  051117  042503  040440
 7491  045332  050040  051101  052111
 7492  045340  020131  051105  047522
 7493  045346  026122  052440  044523
 7494  045354  043516     040
 7495  045357     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 7496  045364  044501  052116  047105
 7497  045372  047101  042503  051040
 7498  045400  043505  051511  042524
 7499  045406  026122     200
 7500  045411     101  020124  044124          .ASCII  'AT THE MAIN MEMORY EVEN WORD, LOW BYTE, PARITY '
 7501  045416  020105  040515  047111
 7502  045424  046440  046505  051117
 7503  045432  020131  053105  047105
 7504  045440  053440  051117  026104
 7505  045446  046040  053517  041040
 7506  045454  052131  026105  050040
 7507  045462  051101  052111  020131
 7508  045470  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
```

```
 7509  045476  026122  020200  042522
 7510  045504  042101  047111  020107
 7511  045512  020101  040504  040524
 7512  045520  050040  052101  042524
 7513  045526  047122  053440  044510
 7514  045534  044103     040
 7515  045537     123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7516  045544  020104  040510  042526
 7517  045552  041440  052501  042523
 7518  045560  020104  047101  042440
 7519  045566  051122  051117  000056
 7520
 7521  045574                              EM141:
 7522  045574  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 7523  045602  046505  051117  020131
 7524  045610  040504  040524  050040
 7525  045616  051101  052111  020131
 7526  045624  044103  041505  042513
 7527  045632  051522  052040  051505
 7528  045640  020124  040506  046111
 7529  045646  042105     056
 7530  045651     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 7531  045656  042514  052040  020117
 7532  045664  047506  041522  020105
 7533  045672  020101  040520  044522
 7534  045700  054524  042440  051122
 7535  045706  051117  020054  051525
 7536  045714  047111  020107
 7537  045720  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 7538  045726  047111  042524  040516
 7539  045734  041516  020105  042522
 7540  045742  044507  052123  051105
 7541  045750  100054
 7542  045752  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY ODD WORD, LOW BYTE, PARITY '
 7543  045760  046440  044501  020116
 7544  045766  042515  047515  054522
 7545  045774  047440  042104  053440
 7546  046002  051117  026104  046040
 7547  046010  053517  041040  052131
 7548  046016  026105  050040  051101
 7549  046024  052111  020131
 7550  046030  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 7551  046036  026122  020200  042522
 7552  046044  042101  047111  020107
 7553  046052  020101  040504  040524
 7554  046060  050040  052101  042524
 7555  046066  047122  053440  044510
 7556  046074  044103     040
 7557  046077     123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7558  046104  020104  040510  042526
 7559  046112  041440  052501  042523
 7560  046120  020104  047101  042440
 7561  046126  051122  051117  000056
 7562
 7563  046134                              EM142:
 7564  046134  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
```

```
 7565  046142  046505  051117  020131
 7566  046150  040504  040524  050040
 7567  046156  051101  052111  020131
 7568  046164  044103  041505  042513
 7569  046172  051522  052040  051505
 7570  046200  020124  040506  046111
 7571  046206  042105     056
 7572  046211     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 7573  046216  042514  052040  020117
 7574  046224  047506  041522  020105
 7575  046232  020101  040520  044522
 7576  046240  054524  042440  051122
 7577  046246  051117  020054  051525
 7578  046254  047111  020107
 7579  046260  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 7580  046266  047111  042524  040516
 7581  046274  041516  020105  042522
 7582  046302  044507  052123  051105
 7583  046310  100054
 7584  046312  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY EVEN WORD, HIGH BYTE, PARITY '
 7585  046320  046440  044501  020116
 7586  046326  042515  047515  054522
 7587  046334  042440  042526  020116
 7588  046342  047527  042122  020054
 7589  046350  044510  044107  041040
 7590  046356  052131  026105  050040
 7591  046364  051101  052111  020131
 7592  046372  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 7593  046400  026122  020200  042522
 7594  046406  042101  047111  020107
 7595  046414  020101  040504  040524
 7596  046422  050040  052101  042524
 7597  046430  047122  053440  044510
 7598  046436  044103     040
 7599  046441     123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7600  046446  020104  040510  042526
 7601  046454  041440  052501  042523
 7602  046462  020104  047101  042440
 7603  046470  051122  051117  000056
 7604
 7605  046476                          EM143:
 7606  046476  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 7607  046504  046505  051117  020131
 7608  046512  040504  040524  050040
 7609  046520  051101  052111  020131
 7610  046526  044103  041505  042513
 7611  046534  051522  052040  051505
 7612  046542  020124  040506  046111
 7613  046550  042105     056
 7614  046553     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR. USING '
 7615  046560  042514  052040  020117
 7616  046566  047506  041522  020105
 7617  046574  020101  040520  044522
 7618  046602  054524  042440  051122
 7619  046610  051117  020054  051525
 7620  046616  047111  020107
```

```
 7621  046622  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 7622  046630  047111  042524  040516
 7623  046636  041516  020105  042522
 7624  046644  044507  052123  051105
 7625  046652  100054
 7626  046654  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY ODD WORD, HIGH BYTE, PARITY '
 7627  046662  046440  044501  020116
 7628  046670  042515  047515  054522
 7629  046676  047440  042104  053440
 7630  046704  051117  026104  044040
 7631  046712  043511  020110  054502
 7632  046720  042524  020054  040520
 7633  046726  044522  054524     040
 7634  046733     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 7635  046740  051105  100054  051040
 7636  046746  040505  044504  043516
 7637  046754  040440  042040  052101
 7638  046762  020101  040520  052124
 7639  046770  051105  020116  044127
 7640  046776  041511  020110
 7641  047002  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7642  047010  044040  053101  020105
 7643  047016  040503  051525  042105
 7644  047024  040440  020116  051105
 7645  047032  047522  027122     000
 7646
 7647  047037     040  052040  051505  DH140:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'DATA.'<TAB>'ADDRESS.'
 7648  047044  027124  041411  046101
 7649  047052  020114  052101  050040
 7650  047060  027103  042011  052101
 7651  047066  027101  040411  042104
 7652  047074  042522  051523  000056
 7653
 7654          047037                  DH141=DH140
 7655
 7656          047037                  DH142=DH140
 7657
 7658          047037                  DH143=DH140
 7659
 7660  047102     004       003     000  DF140:  .BYTE   4,3,0,2
 7661  047105     002
 7662
 7663          047102                  DF141=DF140
 7664
 7665          047102                  DF142=DF140
 7666
 7667          047102                  DF143=DF140
 7668
 7669                                          .EVEN
 7670  047106  001224  001116  001230  DT140:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
 7671  047114  001232  000000
 7672
 7673          047106                  DT141=DT140
 7674
 7675          047106                  DT142=DT140
 7676
```

```
 7677            047106                  DT143=DT140
 7678
 7679
 7680  047120  051105  047522  020122    EM132:  .ASCII  'ERROR REGISTER TEST WAS UNABLE TO CAUSE A TIME OUT,'
 7681  047126  042522  044507  052123
 7682  047134  051105  052040  051505
 7683  047142  020124  040527  020123
 7684  047150  047125  041101  042514
 7685  047156  052040  020117  040503
 7686  047164  051525  020105  020101
 7687  047172  044524  042515  047440
 7688  047200  052125     054
 7689  047203     200  052101  040440            .ASCIZ  <CRLF>'AT AN ADDRESS WHICH SHOULD HAVE TIMED OUT.'
 7690  047210  020116  042101  051104
 7691  047216  051505  020123  044127
 7692  047224  041511  020110  044123
 7693  047232  052517  042114  044040
 7694  047240  053101  020105  044524
 7695  047246  042515  020104  052517
 7696  047254  027124     000
 7697
 7698  047257     105  051122  051117    EM133:  .ASCII  'ERROR REGISTER TEST FAILED.'
 7699  047264  051040  043505  051511
 7700  047272  042524  020122  042524
 7701  047300  052123  043040  044501
 7702  047306  042514  027104
 7703  047312  040600  052106  051105            .ASCII  <CRLF>'AFTER CAUSING A TIME OUT THE ERROR REGISTER SHOULD '
 7704  047320  041440  052501  044523
 7705  047326  043516  040440  052040
 7706  047334  046511  020105  052517
 7707  047342  020124  044124  020105
 7708  047350  051105  047522  020122
 7709  047356  042522  044507  052123
 7710  047364  051105  051440  047510
 7711  047372  046125  020104
 7712  047376  040510  042526  041040            .ASCIZ  'HAVE BEEN SET TO : 000000.'
 7713  047404  042505  020116  042523
 7714  047412  020124  047524  035040
 7715  047420  030040  030060  030060
 7716  047426  027060     000
 7717
 7718  047431     103  047117  051124    EM134:  .ASCII  'CONTROL REGISTER, DISABLE TRAPS, TEST FAILED.'
 7719  047436  046117  051040  043505
 7720  047444  051511  042524  026122
 7721  047452  042040  051511  041101
 7722  047460  042514  052040  040522
 7723  047466  051520  020054  042524
 7724  047474  052123  043040  044501
 7725  047502  042514  027104
 7726  047506  040600  052040  040522            .ASCIZ  <CRLF>'A TRAP OCCURRED WITH BIT 0 SET IN THE CONTROL REGISTER.'
 7727  047514  020120  041517  052503
 7728  047522  051122  042105  053440
 7729  047530  052111  020110  044502
 7730  047536  020124  020060  042523
 7731  047544  020124  047111  052040
 7732  047552  042510  041440  047117
```

```
 7733  047560  051124  046117  051040
 7734  047566  043505  051511  042524
 7735  047574  027122     000
 7736
 7737  047577     105  051122  051117  EM135:  .ASCII  'ERROR REGISTER, LOCK UP, TEST FAILED.'
 7738  047604  051040  043505  051511
 7739  047612  042524  026122  046040
 7740  047620  041517  020113  050125
 7741  047626  020054  042524  052123
 7742  047634  043040  044501  042514
 7743  047642  027104
 7744  047644  040600  052106  051105          .ASCII  <CRLF>'AFTER FORCING MULTIPLE ERRORS, TWO, THE ERROR '
 7745  047652  043040  051117  044503
 7746  047660  043516  046440  046125
 7747  047666  044524  046120  020105
 7748  047674  051105  047522  051522
 7749  047702  020054  053524  026117
 7750  047710  052040  042510  042440
 7751  047716  051122  051117     040
 7752  047723     122  043505  051511          .ASCIZ  'REGISTERS WAS INSORRECTLY SET.'
 7753  047730  042524  051522  053440
 7754  047736  051501  044440  051516
 7755  047744  051117  042522  052103
 7756  047752  054514  051440  052105
 7757  047760  000056
 7758
 7759  047762  052600  042516  050130  EM150:  .ASCIZ  <CRLF>'UNEXPECTED CPU ERROR TRAPPED TO VECTOR ERRVEC (4)!'
 7760  047770  041505  042524  020104
 7761  047776  050103  020125  051105
 7762  050004  047522  020122  051124
 7763  050012  050101  042520  020104
 7764  050020  047524  053040  041505
 7765  050026  047524  020122  051105
 7766  050034  053122  041505  024040
 7767  050042  024464  000041
 7768
 7769                          ;THESE ARE DATA HEADERS:
 7770
 7771  050046  020040  042524  052123  DH1:    .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'PHYSICAL ADDR.'<TAB>'CALL AT PC.'
 7772  050054  004456  043440  047522
 7773  050062  050125  004456  044120
 7774  050070  051531  041511  046101
 7775  050076  040440  042104  027122
 7776  050104  041411  046101  020114
 7777  050112  052101  050040  027103
 7778  050120     000
 7779  050121     040  052040  051505  DH14:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADDR REG.'
 7780  050126  027124  041411  046101
 7781  050134  020114  052101  050040
 7782  050142  027103  042411  051122
 7783  050150  051117  040440  042104
 7784  050156  020122  042522  027107
 7785  050164  052011  040522  020120          .ASCII  <TAB>'TRAP AT PC.'<TAB>
 7786  050172  052101  050040  027103
 7787  050200     011
 7788  050201     105  051122  051117          .ASCIZ  'ERROR REG.'
```

```
 7789  050206  051040  043505  000056
 7790
 7791  050214  020040  042524  052123  DH15:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
 7792  050222  004456  040503  046114
 7793  050230  040440  020124  041520
 7794  050236  000056
 7795
 7796  050240  020040  042524  052123  DH55:   .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'REG ADDRESS.'
 7797  050246  004456  051124  050101
 7798  050254  040440  020124  041520
 7799  050262  004456  040503  046114
 7800  050270  040440  020124  041520
 7801  050276  004456  042522  020107
 7802  050304  042101  051104  051505
 7803  050312  027123     000
 7804
 7805          050240                  DH56=DH55
 7806
 7807          050240                  DH57=DH55
 7808
 7809          050240                  DH60=DH55
 7810
 7811          050240                  DH61=DH55
 7812
 7813          050240                  DH62=DH55
 7814
 7815  050315     040  052040  051505  DH63:   .ASCII  '  TEST."<TAB>'CALL AT PC.'<TAB>'CONTROL.'
 7816  050322  027124  041411  046101
 7817  050330  020114  052101  050040
 7818  050336  027103  041411  047117
 7819  050344  051124  046117     056
 7820  050351     115  044501  052116          .ASCIZ  'MAINT.'<TAB>'(DATA READ FROM EACH REGISTER)'
 7821  050356  004456  042050  052101
 7822  050364  020101  042522  042101
 7823  050372  043040  047522  020115
 7824  050400  040505  044103  051040
 7825  050406  043505  051511  042524
 7826  050414  024522     000
 7827
 7828  050417     040  052040  051505  DH64:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'CONTROL REGISTER DATA.'
 7829  050424  027124  041411  046101
 7830  050432  020114  052101  050040
 7831  050440  027103  041411  047117
 7832  050446  051124  046117  051040
 7833  050454  043505  051511  042524
 7834  050462  020122  040504  040524
 7835  050470  000056
 7836
 7837  050472  020040  042524  052123  DH65:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'LOW ORD.'<TAB>'HIGH ORD.'
 7838  050500  004456  040503  046114
 7839  050506  040440  020124  041520
 7840  050514  004456  047514  020127
 7841  050522  051117  027104  044011
 7842  050530  043511  020110  051117
 7843  050536  027104
 7844  050540  024011  040504  040524          .ASCIZ  <TAB>'(DATA READ FROM ADR. REG.)'
```

```
 7845  050546  051040  040505  020104
 7846  050554  051106  046517  040440
 7847  050562  051104  020056  042522
 7848  050570  027107  000051
 7849
 7850  050574  020040  042524  052123  DH66:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>'READ.'
 7851  050602  004456  040503  046114
 7852  050610  040440  020124  041520
 7853  050616  004456  051127  052117
 7854  050624  027105  051011  040505
 7855  050632  027104
 7856  050634  042411  050130  041505          .ASCIZ  <TAB>'EXPECTED.'
 7857  050642  042524  027104     000
 7858
 7859  050647     040  052040  051505  DH67:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN READ FROM THE '
 7860  050654  027124  041411  046101
 7861  050662  020114  052101  050040
 7862  050670  027103  050011  052101
 7863  050676  042524  047122  051040
 7864  050704  040505  020104  051106
 7865  050712  046517  052040  042510
 7866  050720     040
 7867  050721     110  052111  046457          .ASCIZ  'HIT/MISS REGISTER.'
 7868  050726  051511  020123  042522
 7869  050734  044507  052123  051105
 7870  050742  000056
 7871
 7872          050647                  DH70=DH67
 7873
 7874          050647                  DH71=DH67
 7875
 7876          050647                  DH72=DH67
 7877
 7878          050647                  DH73=DH67
 7879
 7880          050647                  DH74=DH67
 7881
 7882  050744  020040  042524  052123  DH75:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>
 7883  050752  004456  040503  046114
 7884  050760  040440  020124  041520
 7885  050766  004456  043440  047522
 7886  050774  050125  004456
 7887  051000  042101  051104  051505          .ASCIZ  'ADDRESS.'<TAB>'PATTERN IN CONTROL REG.'
 7888  051006  027123  050011  052101
 7889  051014  042524  047122  044440
 7890  051022  020116  047503  052116
 7891  051030  047522  020114  042522
 7892  051036  027107     000
 7893
 7894          050744                  DH76=DH75
 7895
 7896  051041     040  052040  051505  DH77:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
 7897  051046  027124  041411  046101
 7898  051054  020114  052101  050040
 7899  051062  027103     000
 7900
```

```
                                                DH117=DH75
051065    040  052040  051505  DH120:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN IN CONTROL REG.'
051072 027124  041411  046101
051100 020114  052101  050040
051106 027103  050011  052101
051114 042524  047122  044440
051122 020116  047503  052116
051130 047522  020114  042522
051136 027107     000
051141    040  052040  051505  DH121:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TEST ADDRESS.'
051146 027124  041411  046101
051154 020114  052101  050040
051162 027103  052011  051505
051170 020124  042101  051104
051176 051505  027123     000
051203    040  052040  051505  DH122:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>
051210 027124  041411  046101
051216 020114  052101  050040
051224 027103  053411  047522
051232 042524  004456
051236 044124  047105  041440          .ASCIZ  'THEN CLEARED AND READ.'
051244 042514  051101  042105
051252 040440  042116  051040
051260 040505  027104     000
051265    040  042524  052123  DH123:  .ASCIZ  ' TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>'READ.'
051272 004456  040503  046114
051300 040440  020124  041520
051306 004456  051127  052117
051314 027105  051011  040505
051322 027104     000
                                                DH124=DH14
051325    040  052040  051505  DH125:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ADDRESS.'
051332 027124  041411  046101
051340 020114  052101  050040
051346 027103  040411  042104
051354 042522  051523  000056
051362 020040  042524  052123  DH126:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'
051370 004456  040503  046114
051376 040440  020124  041520
051404 004456  051124  050101
051412 040440  020124  041520
051420    056
051421    011  051105  047522          .ASCIZ  <TAB>'ERROR REG.'
051426 020122  042522  027107
051434    000
051435    040  052040  051505  DH127:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN USED.'
051442 027124  041411  046101
```

```
 7957  051450  020114  052101  050040
 7958  051456  027103  050011  052101
 7959  051464  042524  047122  052440
 7960  051472  042523  027104     000
 7961
 7962  051477     040  052040  051505  DH130:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADR REG.'
 7963  051504  027124  041411  046101
 7964  051512  020114  052101  050040
 7965  051520  027103  042411  051122
 7966  051526  051117  040440  051104
 7967  051534  051040  043505     056
 7968  051541     011  051105  047522          .ASCIZ  <TAB>'ERROR REG.'
 7969  051546  020122  042522  027107
 7970  051554     000
 7971
 7972  051555     040  052040  051505  DH131:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
 7973  051562  027124  041411  046101
 7974  051570  020114  052101  050040
 7975  051576  027103  052011  040522
 7976  051604  020120  052101  050040
 7977  051612  027103     011
 7978  051615     105  051122  051117          .ASCIZ  'ERROR ADR REG.'
 7979  051622  040440  051104  051040
 7980  051630  043505  000056
 7981
 7982          051325                  DH132=DH125
 7983
 7984          051362                  DH133=DH126
 7985
 7986  051634  020040  042524  052123  DH134:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
 7987  051642  004456  040503  046114
 7988  051650  040440  020124  041520
 7989  051656  004456  051124  050101
 7990  051664  040440  020124  041520
 7991  051672  004456
 7992  051674  047503  052116  047522          .ASCIZ  'CONTROL REG.'
 7993  051702  020114  042522  027107
 7994  051710     000
 7995
 7996          051041                  DH135=DH77
 7997
 7998  051711     040  052040  051505  DH150:  .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'CPU ERROR REGISTER.'
 7999  051716  027124  052011  040522
 8000  051724  020120  052101  050040
 8001  051732  027103  041411  046101
 8002  051740  020114  052101  050040
 8003  051746  027103  041411  052520
 8004  051754  042440  051122  051117
 8005  051762  051040  043505  051511
 8006  051770  042524  027122     000
 8007                                  ;THESE ARE DATA FORMAT DESIGNATORS FOR THE DATA TABLE:
 8008
 8009  051775     004     004     003  DF1:    .BYTE   4,4,3,3
 8010  052000     003
 8011
 8012  052001     004     003     007  DF14:   .BYTE   4,3,7,3,0
```

```
 8013  052004      003      000
 8014
 8015  052006      004      003          DF15:   .BYTE   4,3
 8016
 8017  052010      004      003      003 DF55:   .BYTE   4,3,3,2
 8018  052013      002
 8019
 8020              052010                DF56=DF55
 8021
 8022              052010                DF57=DF55
 8023
 8024              052010                DF60=DF55
 8025
 8026              052010                DF61=DF55
 8027
 8028              052010                DF62=DF55
 8029
 8030  052014      004      003      000 DF63:   .BYTE   4,3,0,0,0
 8031  052017      000      000
 8032
 8033              052014                DF64=DF63
 8034
 8035              052014                DF65=DF63
 8036
 8037              052014                DF66=DF63
 8038
 8039              052014                DF67=DF63
 8040
 8041              052014                DF70=DF63
 8042
 8043              052014                DF71=DF63
 8044
 8045              052014                DF72=DF63
 8046
 8047              052014                DF73=DF63
 8048
 8049              052014                DF74=DF63
 8050
 8051  052021      004      003      004 DF75:   .BYTE   4,3,4,2,0
 8052  052024      002      000
 8053
 8054              052021                DF76=DF75
 8055
 8056  052026      004      003      005 DF77:   .BYTE   4,3,5,2,5,0,5,2,5,0
 8057  052031      002      005      000
 8058  052034      005      002      005
 8059  052037      000
 8060
 8061
 8062              052021                DF117=DF75
 8063
 8064  052040      004      003      000 DF120:  .BYTE   4,3,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0
 8065  052043      005      000      005
 8066  052046      000      005      000
 8067  052051      005      000      005
 8068  052054      000      005      000
```

```
 8069  052057    005     000     005
 8070  052062    000     005     000
 8071  052065    005     000     005
 8072  052070    000     005     000
 8073
 8074  052073    004     003     002  DF121:  .BYTE   4,3,2,2
 8075  052076    002
 8076
 8077  052077    004     003     000  DF122:  .BYTE   4,3,0,0
 8078  052102    000
 8079
 8080            052077               DF123=DF122
 8081
 8082  052103    004     003     007  DF124:  .BYTE   4,3,7,3,0,5,0,
 8083  052106    003     000     005
 8084  052111    000     000
 8085
 8086  052113    004     003     002  DF125:  .BYTE   4,3,2,0
 8087  052116    000
 8088
 8089  052117    004     003     003  DF126:  .BYTE   4,3,3,0,5,2,5,2
 8090  052122    000     005     002
 8091  052125    005     002
 8092
 8093  052127    004     003     000  DF127:  .BYTE   4,3,0
 8094
 8095            052113               DF130=DF125
 8096
 8097  052132    004     003     003  DF131:  .BYTE   4,3,3,2,5,0,5,0,5,0
 8098  052135    002     005     000
 8099  052140    005     000     005
 8100  052143    000
 8101
 8102            052113               DF132=DF125
 8103
 8104            052117               DF133=DF126
 8105
 8106  052144    004     003     003  DF134:  .BYTE   4,3,3,0,5,2,0
 8107  052147    000     005     002
 8108  052152    000
 8109
 8110  052153    004     003     005  DF135:  .BYTE   4,3,5,0,5,0,5,2,5,2
 8111  052156    000     005     000
 8112  052161    005     002     005
 8113  052164    002
 8114
 8115  052165    004     003     003  DF150:  .BYTE   4,3,3,0
 8116  052170    000
 8117
 8118            052172               .EVEN
 8119
 8120                                 ;THESE ARE DATA TABLES:
 8121
 8122  052172  001224  001226  001230  DT1:    .WORD   $TMP0,$TMP1,$TMP2,$ERRPC,0
 8123  052200  001116  000000
 8124
```

```
 8125  052204  001224  001116  001226  DT14:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,0
 8126  052212  001232  001234  000000
 8127
 8128  052220  001224  001226  000000  DT15:   .WORD   $TMP0,$TMP1,0
 8129
 8130
 8131  052226  001224  001226  001116  DT55:   .WORD   $TMP0,$TMP1,$ERRPC,$TMP3,0
 8132  052234  001232  000000
 8133
 8134          052226                  DT56=DT55
 8135
 8136          052226                  DT57=DT55
 8137
 8138          052226                  DT60=DT55
 8139
 8140          052226                  DT61=DT55
 8141
 8142          052226                  DT62=DT55
 8143
 8144  052240  001224  001116  001230  DT63:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
 8145  052246  001232  000000
 8146
 8147  052252  001224  001116  001230  DT64:   .WORD   $TMP0,$ERRPC,$TMP2,0
 8148  052260  000000
 8149
 8150  052262  001224  001116  001230  DT65:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
 8151  052270  001232  000000
 8152
 8153  052274  001224  001116  001230  DT66:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,$TMP4,0
 8154  052302  001232  001234  000000
 8155
 8156          052252                  DT67=DT64
 8157
 8158          052252                  DT70=DT64
 8159
 8160          052252                  DT71=DT64
 8161
 8162          052252                  DT72=DT64
 8163
 8164          052252                  DT73=DT64
 8165
 8166          052252                  DT74=DT64
 8167
 8168  052310  001224  001116  001230  DT75:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP10,$TMP3,0
 8169  052316  001244  001232  000000
 8170
 8171  052324  001224  001116  001230  DT76:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP12,$TMP3,0
 8172  052332  001250  001232  000000
 8173
 8174  052340  001224  001116  035100  DT77:   .WORD   $TMP0,$ERRPC,MTA77,$TMP10,MTB77,$TMP2,MTC77
 8175  052346  001244  035114  001230
 8176  052354  035156
 8177  052356  001250  035213  001232          .WORD   $TMP12,MTD77,$TMP3,0
 8178  052364  000000
 8179
 8180          052324                  DT117=DT76
```

```
8181
8182  052366  001224  001116  001230  DT120:  .WORD   $TMP0,$ERRPC,$TMP2,MTA120,KCR0,MTG120,KCE0
8183  052374  035363  007640  035603
8184  052402  007654
8185  052404  035413  007642  035603          .WORD   MTB120,KCR1,MTG120,KCE1
8186  052412  007656
8187  052414  035443  007644  035603          .WORD   MTC120,KCR2,MTG120,KCE2
8188  052422  007660
8189  052424  035473  007646  035603          .WORD   MTD120,KCR3,MTG120,KCE3
8190  052432  007662
8191  052434  035523  007650  035603          .WORD   MTE120,KCR4,MTG120,KCE4
8192  052442  007664
8193  052444  035553  007652  035603          .WORD   MTF120,KCR5,MTG120,KCE5,0
8194  052452  007666  000000
8195
8196  052456  001224  001116  001230  DT121:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP4,0
8197  052464  001234  000000
8198
8199  052470  001224  001116  001230  DT122:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
8200  052476  001232  000000
8201
8202          052470                  DT123=DT122
8203
8204  052502  001224  001116  001226  DT124:  .WORD   $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,MTA124,$TMP6,0
8205  052510  001232  001234  035644
8206  052516  001240  000000
8207
8208  052522  001224  001116  001230  DT125:  .WORD   $TMP0,$ERRPC,$TMP2,0
8209  052530  000000
8210
8211  052532  001224  001116  001230  DT126:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP7,MTA126,$TMP5,MTB126,$TMP3,0
8212  052540  001242  035736  001236
8213  052546  035764  001232  000000
8214
8215          052522                  DT127=DT125
8216
8217  052554  001224  001116  001230  DT130:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP4,0
8218  052562  001234  000000
8219
8220  052566  001224  001116  001230  DT131:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA131,$TMP5
8221  052574  001232  036016  001236
8222  052602  036100  001240  036133          .WORD   MTB131,$TMP6,MTC131,$TMP7,0
8223  052610  001242  000000
8224
8225          052522                  DT132=DT125
8226
8227          052532                  DT133=DT126
8228
8229  052614  001224  001116  001230  DT134:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA134,$TMP4,$TMP6,0
8230  052622  001232  036161  001234
8231  052630  001240  000000
8232
8233  052634  001224  001116  036215  DT135:  .WORD   $TMP0,$ERRPC,MTA135,$TMP2,MTB135,$TMP3
8234  052642  001230  036245  001232
8235  052650  036267  001234  036323          .WORD   MTC135,$TMP4,MTD135,$TMP6,0
8236  052656  001240  000000
```

```
 8237
 8238  052662  001224  001226  001230  DT150:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
 8239  052670  001232  000000
 8240
 8241  052674  000000  000000  000000  BOTTOM: .WORD   0,0,0
 8242          060702                  .=.+6000
 8243  060702                          BOTPRG:
 8244          000001                  .END
```

```
ABORTT  032052          5907    6124#
ADRNG   034355          6170    6601#
BACKAD  032170          6140*   6149    6150#
BIT0   = 000001           137#   972     973     980    1333    1424    1522    4486    4560    4625    5231    5352    6040
BIT00  = 000001           127#   137
BIT01  = 000002           126#   136
BIT02  = 000004           125#   135
BIT03  = 000010           124#   134
BIT04  = 000020           123#   133
BIT05  = 000040           122#   132
BIT06  = 000100           121#   131
BIT07  = 000200           120#   130     1031
BIT08  = 000400           119#   129     5516
BIT09  = 001000           118#   128     5527    5593
BIT1   = 000002           136#  1429     1527
BIT10  = 002000           117#  5576
BIT11  = 004000           116#   975     976     978    5534
BIT12  = 010000           115#
BIT13  = 020000           114#  5583
BIT14  = 040000           113#   967     968     970
BIT15  = 100000           112#   981     982     984     986     987     989    6043
BIT2   = 000004           135#  1434     1532
BIT3   = 000010           134#
BIT4   = 000020           133#
BIT5   = 000040           132#
BIT6   = 000100           131#  1089     6266
BIT7   = 000200           130#  6245
BIT8   = 000400           129#  1002
BIT9   = 001000           128#  1329     5518
BOTPRG  060702           8243#
BOTTOM  052674           1079    6251    8241#
BPTVEC= 000014             144#
CACHVE= 000114             151#  1092*   1109*   1201*   1566*   1659*   1750*   1962*   2073*   2141*   2157*   2184*   2260*
                          2345*  2429*   2516*   2597*   2692*   2780*   2879*   2978*   3077*   3176*   3275*   3374*   3473*
                          3576*  3700*   3831*   3943*   4054*   4165*   4276*   4368*   4487*   4561*   4626*   4686*   4710*
                          4824*  4850*   4966*   4991*   5107*   5133*   5224*   5235*   5257*   5345*   5356*   5378*   6104*
                          6110*  6138*
CHAINQ  032414           5471    6250#
CISP    001313            561#
CLEAN   032102           5906    6136#
CNRNG   034565           6184    6628#
CONCMS  033330           6249    6481#
CONFLG  032316           1164*   1226*   1233*   1353*   1936*   6180    6206#
CONFL2  032332           1445*   1541*   1604*   1627*   1697*   1720*   1909*   1934    6212#
CONTRL= 177746             161#   964     994*    995    1162    1207*   1209    1315    1377*   1392*   1405*   1436*   1474*
                          1490*  1503*   1533*   1582*   1595*   1606*   1619*   1629*   1675*   1688*   1699*   1712*   1722*
                          1765*  1767*   1774*   1913*   1976*   1994*   2010*   2081*   2156*   2188*   2263*   2348*   2432*
                          2519*  2600*   2693*   2783*   2882*   2981*   3080*   3179*   3278*   3377*   3476*   3609*   3725*
                          3832*  3944*   4055*   4166*   4277*   4485*   4486*   4513    4545*   4560*   4578    4610*   4625*
                          4643   4683*   4821*   4961*   5102*   5226*   5347*   5518*   6145*
CPSPUR  031726           1091    3577    4423    6096#   6139
CPUERR= 177766            174#  1134    1182*   1185*   4421    5601*   6098    6147*
CR     = 000015            49#  5714    5724
CRLF   = 000200            50#   933    5687    5724    6481    6483    6490    6500    6503    6510    6520    6530    6533
                          6543   6549    6552    6557    6564    6569    6576    6586    6589    6596    6601    6615    6628
                          6640   6654    6667    6677    6690    6695    6703    6709    6715    6720    6726    6732    6745
```

```
                    6757    6762    6768    6778    6784    6789    6795    6806    6816    6822    6824    6826    6831
                    6847    6861    6880    6899    6915    6931    6948    6955    6974    6982    7001    7009    7015
                    7033    7041    7078    7086    7104    7113    7137    7146    7179    7198    7207    7232    7241
                    7257    7277    7294    7317    7346    7368    7385    7407    7423    7461    7467    7488    7495
                    7508    7530    7537    7550    7572    7579    7592    7614    7621    7634    7689    7703    7726
                    7744    7759
DF1       051775     587    8009#
DF117 =   052021     825    8062#
DF120     052040     828    8064#
DF121     052073     831    8074#
DF122     052077     834    8077#   8080
DF123 =   052077     837    8080#
DF124     052103     840    8082#
DF125     052113    8086#   8095    8102
DF126     052117    8089#   8104
DF127     052127     849    8093#
DF130 =   052113     852    8095#
DF131     052132     856    8097#
DF132 =   052113     859    8102#
DF133 =   052117     862    8104#
DF134     052144     865    8106#
DF135     052153     868    8110#
DF14      052001     620    8012#
DF140     047102     877    7660#   7663    7665    7667
DF141 =   047102     880    7663#
DF142 =   047102     883    7665#
DF143 =   047102     886    7667#
DF15      052006     623    8015#
DF150     052165     901    8115#
DF55      052010     721    8017#   8020    8022    8024    8026    8028
DF56  =   052010     724    8020#
DF57  =   052010     727    8022#
DF60  =   052010     730    8024#
DF61  =   052010     733    8026#
DF62  =   052010     736    8028#
DF63      052014     739    8030#   8033    8035    8037    8039    8041    8043    8045    8047    8049
DF64  =   052014     742    8033#
DF65  =   052014     745    8035#
DF66  =   052014     748    8037#
DF67  =   052014     751    8039#
DF70  =   052014     754    8041#
DF71  =   052014     757    8043#
DF72  =   052014     760    8045#
DF73  =   052014     763    8047#
DF74  =   052014     766    8049#
DF75      052021     770    8051#   8054    8062
DF76  =   052021     773    8054#
DF77      052026     776    8056#
DH1       050046     587    7771#
DH117 =   050744     825    7902#
DH120     051065     828    7904#
DH121     051141     831    7913#
DH122     051203     834    7920#
DH123     051265     837    7930#
DH124 =   050121     840    7937#
DH125     051325    7939#   7982
```

```
DH126    051362          7945#   7984
DH127    051435           849    7955#
DH130    051477           852    7962#
DH131    051555           856    7972#
DH132 =  051325           859    7982#
DH133 =  051362           862    7984#
DH134    051634           865    7986#
DH135 =  051041           868    7996#
DH14     050121           620    7779#   7937
DH140    047037           877    7647#   7654    7656    7658
DH141 =  047037           880    7654#
DH142 =  047037           883    7656#
DH143 =  047037           886    7658#
DH15     050214           623    7791#
DH150    051711           901    7998#
DH55     050240           721    7796#   7805    7807    7809    7811    7813
DH56  =  050240           724    7805#
DH57  =  050240           727    7807#
DH60  =  050240           730    7809#
DH61  =  050240           733    7811#
DH62  =  050240           736    7813#
DH63     050315           739    7815#
DH64     050417           742    7828#
DH65     050472           745    7837#
DH66     050574           748    7850#
DH67     050647           751    7859#   7872    7874    7876    7878    7880
DH70  =  050647           754    7872#
DH71  =  050647           757    7874#
DH72  =  050647           760    7876#
DH73  =  050647           763    7878#
DH74  =  050647           766    7880#
DH75     050744           770    7882#   7894    7902
DH76  =  050744           773    7894#
DH77     051041           776    7896#   7996
DISPLA=  177570            44#   5548*   5572*
DT1      052172           587    8122#
DT117 =  052324           825    8180#
DT120    052366           828    8182#
DT121    052456           831    8196#
DT122    052470           834    8199#   8202
DT123 =  052470           837    8202#
DT124    052502           840    8204#
DT125    052522          8208#   8215    8225
DT126    052532          8211#   8227
DT127 =  052522           849    8215#
DT130    052554           852    8217#
DT131    052566           856    8220#
DT132 =  052522           859    8225#
DT133 =  052532           862    8227#
DT134    052614           865    8229#
DT135    052634           868    8233#
DT14     052204           620    8125#
DT140    047106           877    7670#   7673    7675    7677
DT141 =  047106           880    7673#
DT142 =  047106           883    7675#
DT143 =  047106           886    7677#
```

DT15	052220	623	8128#						
DT150	052662	901	8238#						
DT55	052226	721	8131#	8134	8136	8138	8140	8142	
DT56 =	052226	724	8134#						
DT57 =	052226	727	8136#						
DT60 =	052226	730	8138#						
DT61 =	052226	733	8140#						
DT62 =	052226	736	8142#						
DT63	052240	739	8144#						
DT64	052252	742	8147#	8156	8158	8160	8162	8164	8166
DT65	052262	745	8150#						
DT66	052274	748	8153#						
DT67 =	052252	751	8156#						
DT70 =	052252	754	8158#						
DT71 =	052252	757	8160#						
DT72 =	052252	760	8162#						
DT73 =	052252	763	8164#						
DT74 =	052252	766	8166#						
DT75	052310	770	8168#						
DT76	052324	773	8171#	8180					
DT77	052340	776	8174#						
EMTVEC=	000030	147#	914*	915*					
EM1	036474	587	6836#						
EM117	043250	825	7294#						
EM120	043377	828	7310#						
EM121	043612	831	7336#						
EM122	044013	834	7360#						
EM123	044143	837	7376#						
EM124	044344	840	7399#						
EM127	044552	849	7423#						
EM130	044734	852	7444#						
EM131	045006	856	7452#						
EM132	047120	859	7680#						
EM133	047257	862	7698#						
EM134	047431	865	7718#						
EM135	047577	868	7737#						
EM14	036561	620	6847#						
EM140	045233	877	7479#						
EM141	045574	880	7521#						
EM142	046134	883	7563#						
EM143	046476	886	7605#						
EM15	036620	623	6854#						
EM150	047762	901	7759#						
EM55	036670	721	6861#						
EM56	037034	724	6880#						
EM57	037201	727	6899#						
EM60	037323	730	6915#						
EM61	037447	733	6931#						
EM62	037577	736	6948#						
EM63	037725	739	6965#						
EM64	040144	742	6992#						
EM65	040343	745	7015#						
EM66	040726	748	7060#						
EM67	041010	751	7070#						
EM70	041225	754	7095#						
EM71	041503	757	7128#						

```
EM72    041761              760    7161#
EM73    042203              763    7189#
EM74    042467              766    7223#
EM75    042753              770    7257#  7275
EM76  = 042753              773    7275#
EM77    043112              776    7277#
ENDKB   003520             1005    1010#
ERRNG   034465             6177    6615#
ERRVEC= 000004              140#   1091*  1114   1115*  1128*  1136*  1137   3577*  4398*  5507   5508*  5510*  5513*
                           5982    5983   5987*  5992*  6004*  6009*  6019*  6031*  6032*  6139*
ERTYPE  032616             5585    6127   6324#
ERT1    032730             6356#   6420
ERT2    033142             6360    6365   6375   6383   6389   6401   6410   6414#
ERT3    033146             6395    6417#
ERT4    033156             6354    6419   6422#
ERT5    033160             6334    6423#
GNS   = ****** U            444     932   1039   1043   1047   1056   1064   5898   5899   5900   5901   5902   5903
                           5904    5906   5907   5908   5909   5910   5911   5912   5913   5914
HIADRS= 177742              159#   1150   1266   2147   2212   2220   2228   2297   2306   2314   2381   2390   2398
                           2468    2477   2485   2554   2563   2571   2642   2651   2659   2728   2737   2745   2827
                           2836    2844   2926   2935   2943   3025   3034   3042   3124   3133   3141   3223   3232
                           3240    3322   3331   3339   3421   3430   3438   3520   3529   3537   3645   3663   3671
                           3757    3766   3774   3869   3878   3886   3980   3989   3997   4091   4100   4108   4202
                           4211    4219   4313   4322   4330   4432   4450   4458   4515   4580   4645   4746   4758
                           4767    4886   4898   4907   5027   5039   5048   5169   5181   5190   6113
HIAFLG  032312             1152*  1281*  6166   6204#
HIAFL2  032326             6210#
HIMFLG  032322             1176*  1536*  1939*  6194   6208#
HIMFL2  032336             1439*  1910*  1937   6214#
HITMIS= 177752              163#   1125   1174   1388   1401   1414   1485   1499   1512   1585   1598   1609   1621
                           1678    1691   1702   1714   1786   1984   2000   2030   2791   2890   2989   3088   3187
                           3286    3385   3484   4551   4616
HMRNG   034775             6198    6654#
HT    = 000011              47#    5685   5724
IOTVEC= 000020              145#    912*   913*
JA    = 000001             1104#
JADONE  004622             1129    1185#
JAERR   004400             1115    1134#
JAERR0  004410             1136#   1139
JAERR1  004422             1135    1138#
JAERR2  004470             1145    1150#
JAERR3  004510             1151    1156#
JAERR4  004530             1157    1162#
JAERR5  004550             1163    1168#
JAERR6  004570             1169    1174#
JAERR7  004610             1175    1180#
JAERR9  004612             1148    1154   1160   1166   1172   1178   1182#
JATMP   004376             1114*  1128   1131#  1136
JA1     004346             1118    1120#  1126
JA2     004352             1124#   1138   1183
JA3     004364             1128#
JB    = 000002             1196#
JBDONE  004770             1211    1228   1235#
JBERR1  004724             1222#
JBERR2  004754             1220    1230#
JB1     004666             1206    1207#
```

```
JB2      004710          1217#
JC     = 000003          1252#
JCDONE   005132          1270    1280    1283#
JCERR1   005062          1268    1272#
JC1      005034          1262    1264#
JC2      005054          1269#
JD     = 000004          1239    1298#
JDDONE   005332          1328    1347    1354#
JDERR1   005306          1322    1332    1337    1341    1345    1349#
KA     = 000006          1464#
KADONE   006172          1515    1533#
KAD2     006216          1535    1539#
KAD3     006234          1537    1540    1542#
KAERR1   006122          1487    1519#
KAERR2   006140          1501    1524#
KAERR3   006156          1514    1529#
KAFLG    006120          1473*   1517#   1522*   1527*   1532*   1534    1539
KA1      005720          1474#   1475
KA2      005742          1476    1479#   1480
KA3      005770          1489#   1523
KA4      006014          1491    1493#   1494
KA5      006042          1502#   1528
KA6      006066          1504    1506#   1507
KB     = 000005          1367#   1917
KBDONE   005626          1416    1436#
KBD2     005650          1438    1443#
KBD3     005664          1444    1447#
KBERR1   005556          1389    1421#
KBERR2   005574          1402    1426#
KBERR3   005612          1415    1431#
KBFLG    005554          1376*   1419#   1424*   1429*   1434*   1437    1443
KBTST    003244           956#
KB1      005366          1377#   1378
KB11CM   001312           560#    956*   1004*   1016    1327
KB11E    001310           558#    957*    961*   1000    1002*   1014    1022
KB11EM   001311           559#   1325
KB2      005412          1380    1382#   1383
KB3      005434          1391#   1425
KB4      005460          1393    1395#   1396
KB5      005502          1404#   1430
KB6      005526          1406    1408#   1409
KC     = 000011          1745#
KCCON    007606          1754*   1774    1864*   1867*   1874#   1907
KCDONE   007720          1861    1913#
KCERR    007670          1851    1906#
KCE0     007654          1840    1847    1899#   8182
KCE1     007656          1900#   8185
KCE2     007660          1901#   8187
KCE3     007662          1902#   8189
KCE4     007664          1903#   8191
KCE5     007666          1904#   8193
KCFLG1   007610          1755*   1859*   1876#
KCPTR    007612          1757*   1771    1837    1854*   1855    1878#
KCR0     007640          1829*   1846    1892#   8182
KCR1     007642          1831*   1893#   8185
KCR2     007644          1832*   1894#   8187
```

```
KCR3    007646          1833*   1895#   8189
KCR4    007650          1834*   1896#   8191
KCR5    007652          1835*   1897#   8193
KCTBL   007614          1757    1881#
KCTBLB  007636          1855    1890#
KC0     007164          1756#   1865    1870
KC1     007200          1756    1762#   1857
KC10    007404          1828#
KC11    007444          1837#
KC12    007462          1841#   1844
KC13    007512          1849#   1852
KC14    007522          1850    1852#
KC15    007524          1854#   1911
KC16    007560          1860    1863#
KC17    007574          1863    1867#
KC2     007262          1776#   1784
KC2.5   007270          1775    1777    1779#
KC3     007322          1772    1787    1801#
KC4     007332          1802    1805#
KC5     007342          1806    1809#
KC6     007352          1810    1813#
KC7     007362          1814    1817#
KC8     007372          1818    1821#
KC9     007402          1822    1825#
KD    = 000007          1561#
KDPAR0= 172360           317#
KDPAR1= 172362           318#
KDPAR2= 172364           319#
KDPAR3= 172366           320#
KDPAR4= 172370           321#
KDPAR5= 172372           322#
KDPAR6= 172374           323#
KDPAR7= 172376           324#
KDPDR0= 172320           295#
KDPDR1= 172322           296#
KDPDR2= 172324           297#
KDPDR3= 172326           298#
KDPDR4= 172330           299#
KDPDR5= 172332           300#
KDPDR6= 172334           301#
KDPDR7= 172336           302#
KE    = 000010          1654#   1917
KERSTK= 001100            34#
KF    = 000013          1957#
KFTMP1  010206          2043#
KFTMP2  010210          1966    2044#
KF1     010014          1966#
KF2     010076          1985    1994#
KF3     010134          2001    2010#
KF4     010204          2033    2041#
KF5     010212          2041    2046#
KIPAR0= 172340           306#   3587    3702    3810    3922    4033    4144    4255    4376    4802    4942    5083    5962
                        5996    6457
KIPAR1= 172342           307#
KIPAR2= 172344           308#
KIPAR3= 172346           309#
```

```
KIPAR4= 172350            310#
KIPAR5= 172352            311#
KIPAR6= 172354            312#   4396*
KIPAR7= 172356            313#   6013
KIPDR0= 172300            284#    981*   982    984*   3589   3704   3812   3924   4035   4146   4257   4378   4804
                         4944    5085
KIPDR1= 172302            285#
KIPDR2= 172304            286#
KIPDR3= 172306            287#
KIPDR4= 172310            288#
KIPDR5= 172312            289#
KIPDR6= 172314            290#
KIPDR7= 172316            291#
KTMP1D  006560           1632#
KTMP1E  007110           1725#
KTMP2D  006562           1571    1633#
KTMP2E  007112           1664    1726#
KV    = 000044           4475#
KVDONE  023376           4508    4519#
KVERR   023336           4487    4510#
KV1     023320           4490    4497#
KV2     023326           4500#
KX    = 000045           4535#
KXDONE  023576           4569    4585#
KXERR   023536           4561    4575#
KX1     023500           4552    4560#
KX2     023530           4546    4567#
KY    = 000012           1933#
KY1     007742           1935    1937#
KY2     007756           1938    1940#
KZ    = 000046           4600#
KZDONE  023776           4634    4650#
KZERR   023736           4626    4640#
KZ1     023700           4617    4625#
KZ2     023730           4611    4632#
K1D     006272           1571#
K1E     006622           1664#
K2D     006336           1582#
K2E     006666           1675#
K3D     006376           1586    1593#
K3E     006726           1679    1686#
K4D     006442           1599    1606#
K4E     006772           1692    1699#
K5D     006502           1610    1617#
K5E     007032           1703    1710#
K6D     006552           1616    1622   1629#
K6E     007102           1709    1715   1722#
K7D     006564           1630    1635#
K7E     007114           1723    1728#
LF    = 000012            48#    5718   5724
LOADRS= 177740           158#    1117   1144   1265   2146   2211   2218   2227   2296   2304   2313   2380   2388
                        2397    2467   2475   2484   2553   2561   2570   2641   2649   2658   2727   2735   2744
                        2826    2834   2843   2925   2933   2942   3024   3032   3041   3123   3131   3140   3222
                        3230    3239   3321   3329   3338   3420   3428   3437   3519   3527   3536   3644   3662
                        3669    3756   3764   3773   3868   3876   3885   3979   3987   3996   4090   4098   4107
                        4201    4209   4218   4312   4320   4329   4431   4449   4456   4514   4579   4644   4737
```

```
                    4745   4757   4765   4877   4885   4897   4905   5018   5026   5038   5046   5160   5168
                    5180   5188   5293   5304   5414   5425   6108   6112
LOAFLG  032310      1110   1146*  1278*  6164   6203#
LOAFL2  032324      6209#
LOC   = 027070      1795#  1796#  1797#  1798   2266#  2267#  2268#  2269   2351#  2352#  2353#  2354   2435#
                    2436#  2437#  2438   2522#  2523#  2524#  2525   2610#  2611#  2612#  2613   2696#  2697#
                    2698#  2699   3728#  3729#  3730#  3731   3837#  3838#  3839#  3840   3949#  3950#  3951#
                    3952   4060#  4061#  4062#  4063   4171#  4172#  4173#  4174   4282#  4283#  4284#  4285
                    4492#  4493#  4494#  4495   4691#  4692#  4693#  4694   4715#  4716#  4717#  4718   4829#
                    4830#  4831#  4832   4855#  4856#  4857#  4858   4971#  4972#  4973#  4974   4996#  4997#
                    4998#  4999   5112#  5113#  5114#  5115   5138#  5139#  5140#  5141   5281#  5282#  5283#
                    5284   5402#  5403#  5404#  5405
LOOP    004146      1073#  5480
MA    = 000014      2065#
MAB   = 000015      2174#
MABDON  010770      2200   2216   2222   2237#
MABRR0  010600      2184   2202#
MABRR1  010610      2205#
MABRR2  010626      2210#  2219   2221
MABRR3  010662      2208   2218#
MABRR4  010704      2203   2224#
MABR15  010612      2206#  2235
MAB2    010562      2196#
MADONE  010462      2132   2156#
MAERR   010376      2073   2134#
MAERR1  010420      2139   2145#
MAINT = 177750       162#   962*   965   1168   1217*  1218   1333*  2083   2187   2262   2347   2431   2518
                    2599   2691   2782   2881   2980   3079   3178   3277   3376   3475   3610   3724   3834
                    3946   4057   4168   4279   4489   4564   4629   4688   4712   4826   4852   4968   4993
                    5109   5135   5237   5260   5358   5381   6142*  6241*
MANFLG  032320      1170*  1227*  2119*  4868*  4984*  5125*  5151*  6187   6207#
MANFL2  032334      2128*  2152*  2199*  2233*  2282*  2320*  2366*  2404*  2453*  2491*  2539*  2577*  2627*
                    2665*  2713*  2751*  2812*  2850*  2911*  2949*  3010*  3048*  3109*  3147*  3208*  3246*
                    3307*  3345*  3406*  3444*  3505*  3543*  3629*  3650*  3742*  3780*  3854*  3892*  3965*
                    4003*  4076*  4114*  4187*  4225*  4298*  4336*  4704*  4728*  4842*  5009*  6213#
MAPH0 = 170202       401#
MAPH00= 170202       337#   401    966   4401*
MAPH01= 170206       339#   403
MAPH02= 170212       341#   405
MAPH03= 170216       343#   407
MAPH04= 170222       345#   409
MAPH05= 170226       347#   411
MAPH06= 170232       349#   413
MAPH07= 170236       351#   415
MAPH1 = 170206       403#
MAPH10= 170242       353#
MAPH11= 170246       355#
MAPH12= 170252       357#
MAPH13= 170256       359#
MAPH14= 170262       361#
MAPH15= 170266       363#
MAPH16= 170272       365#
MAPH17= 170276       367#
MAPH2 = 170212       405#
MAPH20= 170302       369#
MAPH21= 170306       371#
```

```
MAPH22= 170312           373#
MAPH23= 170316           375#
MAPH24= 170320           377#
MAPH25= 170326           379#
MAPH26= 170332           381#
MAPH27= 170336           383#
MAPH3 = 170216           407#
MAPH30= 170342           385#
MAPH31= 170346           387#
MAPH32= 170352           389#
MAPH33= 170356           391#
MAPH34= 170362           393#
MAPH35= 170366           395#
MAPH36= 170372           397#
MAPH37= 170376           399#
MAPH4 = 170222           409#
MAPH5 = 170226           411#
MAPH6 = 170232           413#
MAPH7 = 170236           415#
MAPL0 = 170200           400#
MAPL00= 170200           336#    400    3592    3707    3815    3927    4038    4149    4260    4400*    4807    4947    5088
MAPL01= 170204           338#    402
MAPL02= 170210           340#    404
MAPL03= 170214           342#    406
MAPL04= 170220           344#    408
MAPL05= 170224           346#    410
MAPL06= 170230           348#    412
MAPL07= 170234           350#    414
MAPL1 = 170204           402#
MAPL10= 170240           352#
MAPL11= 170244           354#
MAPL12= 170250           356#
MAPL13= 170254           358#
MAPL14= 170260           360#
MAPL15= 170264           362#
MAPL16= 170270           364#
MAPL17= 170274           366#
MAPL2 = 170210           404#
MAPL20= 170300           368#
MAPL21= 170304           370#
MAPL22= 170310           372#
MAPL23= 170314           374#
MAPL24= 170320           376#
MAPL25= 170324           378#
MAPL26= 170330           380#
MAPL27= 170334           382#
MAPL3 = 170214           406#
MAPL30= 170340           384#
MAPL31= 170344           386#
MAPL32= 170350           388#
MAPL33= 170354           390#
MAPL34= 170360           392#
MAPL35= 170364           394#
MAPL36= 170370           396#
MAPL37= 170374           398#
MAPL4 = 170220           408#
```

```
MAPL5 = 170224            410#
MAPL6 = 170230            412#
MAPL7 = 170234            414#
MA1     010276           2085    2088#   2131
MA2     010314           2110#
MA3     010342           2113    2121#
MA4     010366           2122    2130#   2154
MB    = 000016           2250#
MBDONE  011306           2283    2302    2308    2323#
MBERR0  011106           2260    2285#
MB1     011060           2264    2271#
MB2     011064           2273#
MB3     011070           2278#
MC    = 000017           2335#
MCDONE  011622           2367    2386    2392    2407#
MCERR0  011422           2345    2369#
MC1     011374           2349    2356#
MC2     011400           2358#
MC3     011404           2362#
MD    = 000020           2419#
MDDONE  012142           2454    2473    2479    2494#
MDERR0  011742           2429    2456#
MD1     011712           2433    2441#
MD2     011716           2443#
MD3     011724           2449#
ME    = 000021           2506#
MEDONE  012462           2540    2559    2565    2580#
MEERR0  012262           2516    2542#
MEMERR= 177744            160#   1156    1261*   2138    2145    2202    2206*   2207    2213    2231    2285    2291*   2292
                         2298    2317    2369    2375*   2376    2382    2401    2456    2462*   2463    2469    2488    2542
                         2548*   2549    2555    2574    2630    2636*   2637    2643    2662    2716    2722*   2723    2729
                         2748    2815    2821*   2822    2828    2847    2914    2920*   2921    2927    2946    3013    3019*
                         3020    3026    3045    3112    3118*   3119    3125    3144    3211    3217*   3218    3224    3243
                         3310    3316*   3317    3323    3342    3409    3415*   3416    3422    3441    3508    3514*   3515
                         3521    3540    3637    3648    3657*   3658    3664    3745    3751*   3752    3758    3777    3857
                         3863*   3864    3870    3889    3968    3974*   3975    3981    4000    4079    4085*   4086    4092
                         4111    4190    4196*   4197    4203    4222    4301    4307*   4308    4314    4333    4425    4435
                         4444*   4445    4451    4516    4581    4646    4735    4742    4752*   4753    4759    4875    4882
                         4892*   4893    4899    5016    5023    5033*   5034    5040    5158    5165    5175*   5176    5182
                         5290    5297*   5301    5308*   5411    5418*   5422    5429*   5600*   6040    6043    6046*   6105
                         6111    6146*
ME1     012232           2520    2528#
ME2     012236           2530#
ME3     012244           2535#
MF    = 000022           2592#
MFDONE  012776           2628    2647    2653    2668#
MFERR0  012576           2597    2630#
MFPT  = 000007            564#    959
MFPTTR  003512            958    1007#
MF1     012544           2608    2615#
MF2     012556           2601    2621#
MF3     012560           2623#
MG    = 000023           2680#
MGDONE  013316           2714    2733    2739    2754#
MGERR0  013116           2692    2716#
MG1     013064           2694    2701#
```

```
MG2       013072          2704#
MG3       013100          2709#
MH      = 000024          2770#
MHDONE    013662          2813    2832    2838    2853#
MHERR0    013462          2780    2815#
MH1       013442          2786    2802#
MH2       013444          2808#
MI      = 000025          2869#
MIDONE    014226          2912    2931    2937    2952#
MIERR0    014026          2879    2914#
MI1       014006          2885    2901#
MI2       014010          2907#
MJ      = 000026          2968#
MJDONE    014572          3011    3030    3036    3051#
MJERR0    014372          2978    3013#
MJ1       014352          2984    3000#
MJ2       014354          3006#
MK      = 000027          3067#
MKDONE    015136          3110    3129    3135    3150#
MKERR0    014736          3077    3112#
MK1       014716          3083    3099#
MK2       014720          3105#
ML      = 000030          3166#
MLDONE    015502          3209    3228    3234    3249#
MLERR0    015302          3176    3211#
ML1       015262          3182    3198#
ML2       015264          3204#
MMDES     032520          5908    6283#
MMESRS    033334          6261    6483#
MMRFLG    032314          1158*   2215*   2301*   2385*   2472*   2558*   2646*   2732*   2831*   2930*   3029*   3128*   3227*
                          3326*   3425*   3524*   3666*   3761*   3873*   3984*   4095*   4206*   4317*   4453*   4762*   4902*
                          5043*   5185*   6173    6205#
MMRFL2    032330          2234*   2321*   2405*   2492*   2578*   2666*   2752*   2851*   2950*   3049*   3148*   3247*   3346*
                          3445*   3544*   3781*   3893*   4004*   4115*   4226*   4337*   4437*   6211#
MMR0    = 177572           185#    189    2289*   2373*   2460*   2546*   2634*   2720*   2819*   2918*   3017*   3116*   3215*
                          3314*   3413*   3512*   3607*   3655*   3722*   3749*   3830*   3861*   3942*   3972*   4053*   4083*
                          4164*   4194*   4275*   4305*   4392*   4442*   4750*   4849*   4890*   4965*   4988*   5031*   5106*
                          5132*   5173*   6143*   6446
MMR1    = 177574           186#    190
MMR2    = 177576           187#    191
MMR3    = 172516           188#    192    2290*   2374*   2461*   2547*   2635*   2721*   2820*   2919*   3018*   3117*   3216*
                          3315*   3414*   3513*   3606*   3656*   3721*   3750*   3829*   3862*   3941*   3973*   4052*   4084*
                          4163*   4195*   4274*   4306*   4391*   4443*   4751*   4848*   4891*   4964*   4989*   5032*   5105*
                          5131*   5174*   6144*
MMSKIP= 104422            3575    3699    3808    3920    4031    4142    4253    4374    4800    4940    5081    5908#
MMVEC   = 000250           153#
MN      = 000031          3265#
MNRNG     034667          6191    6640#
MO      = 000032          3364#
MODONE    016412          3407    3426    3432    3447#
MOERR0    016212          3374    3409#
MONF      032516          1073*   6255*   6271#
MONTTY    032514          1076*   6262    6269#
MO1       016172          3380    3396#
MO2       016174          3402#
MP      = 000033          3463#
```

```
MPDONE  016756          3506    3525    3531    3546#
MPERR0  016556          3473    3508#
MP1     016536          3479    3495#
MP2     016540          3501#
MQ    = 000043          4363#
MQDONE  023222          4417    4454    4461#
MQERR   023016          4398    4421#
MQVAR   023014          4395*   4396    4419#
MQ1     022774          4413#
MQ2     023032          4422    4425#
MQ3     023120          4426    4440#
MQ4     023122          4438    4442#
MQ5     023146          4448#   4457    4459
MQ6     023202          4446    4456#
MR    = 000034          3565#
MRDONE  017406          3630    3667    3674#
MRERR0  017200          3576    3632#
MR1     017160          3626#
MR2     017212          3634    3637#
MR3     017302          3638    3653#
MR4     017306          3651    3655#
MR5     017332          3661#   3670    3672
MR6     017366          3659    3669#
MS    = 000035          3689#
MSDONE  020026          3743    3762    3768    3783#
MSERR0  017626          3700    3745#
MSG1    036351          1013    6816#
MSG2    036410          1026    6822#
MSG3    036421          1018    6824#
MSG4    036433          1020    6826#
MSG5    036464          1024    6831#
MSIZER  032542          5909    6299#
MS1     017600          3726    3733#
MS2     017604          3735#
MS3     017610          3738#
MT    = 000036          3798#
MTA101  035256          6690#
MTA11   033524          6510#
MTA120  035363          6703#   8182
MTA124  035644          6745#   8204
MTA126  035736          6757#   8211
MTA131  036016          6768#   8220
MTA134  036161          6789#   8229
MTA135  036215          6795#   8233
MTA17   033571          6518#   6541
MTA20   033625          6527#
MTA21   033634          6530#
MTA43   033721          6543#
MTA45   033774          6552#
MTA5    033442          6500#
MTA50   034052          6564#
MTA77   035100          6667#   8174
MTB120  035413          6709#   8185
MTB126  035764          6762#   8211
MTB131  036100          6778#   8222
MTB135  036245          6801#   8233
```

```
MTB17   033576          6520#
MTB21 = 033571          6541#
MTB45   034022          6557#
MTB77   035114          6670#   8174
MTC120  035443          6715#   8187
MTC131  036133          6784#   8222
MTC135  036267          6806#   8235
MTC17   033616          6524#
MTC45   034037          6561#
MTC77   035156          6677#   8174
MTDONE  020452          3855    3874    3880    3895#
MTD120  035473          6720#   8189
MTD135  036323          6812#   8235
MTD77   035213          6683#   8177
MTERR0  020252          3831    3857#
MTE120  035523          6726#   8191
MTF120  035553          6732#   8193
MTG120  035603          6738#   8182    8185    8187    8189    8191    8193
MT1     020222          3835    3843#
MT2     020234          3850#
MU    = 000037          3910#
MUDONE  021072          3966    3985    3991    4006#
MUERR0  020672          3943    3968#
MU1     020646          3947    3955#
MU2     020654          3961#
MV    = 000040          4021#
MVDONE  021512          4077    4096    4102    4117#
MVERR0  021312          4054    4079#
MV1     021266          4058    4066#
MV2     021274          4072#
MW    = 000041          4132#
MWDONE  022132          4188    4207    4213    4228#
MWERR0  021732          4165    4190#
MW1     021706          4169    4177#
MW2     021714          4183#
MX    = 000042          4243#
MXDONE  022552          4299    4318    4324    4339#
MXERR0  022352          4276    4301#
MX1     022326          4280    4288#
MX2     022334          4294#
MOM1  = 000014           433#   1377    2081    2188    2263    2348    2432    2519    2600    2693    3609    3832    4485
                        4683    4821    4961    5102    5226    5347
M1M0  = 000014           432#    433    3725
NA    = 000047          4673#
NADONE  024362          4705    4729    4763    4770#
NA1     024064          4689    4696#
NA10    024342          4754    4765#
NA2     024070          4698#   4737    4743
NA3     024116          4686    4708#
NA4     024140          4713    4720#
NA5     024144          4722#
NA6     024172          4710    4732#
NA7     024216          4736    4740#
NA8     024262          4738    4750#
NA9     024306          4756#   4766    4768
NB    = 000050          4790#
```

```
NBDONE  025052          4843    4869    4903    4910#
NB1     024540          4827    4834#
NB10    025032          4894    4905#
NB2     024544          4836#   4877    4883
NB3     024572          4824    4846#
NB4     024630          4853    4860#
NB5     024634          4862#
NB6     024662          4850    4872#
NB7     024706          4876    4880#
NB8     024752          4878    4890#
NB9     024776          4896#   4906    4908
NC    = 000051          4930#
NCDONE  025552          4985    5010    5044    5051#
NC1     025244          4969    4976#
NC10    025532          5035    5046#
NC2     025250          4978#   5018    5024
NC3     025276          4966    4988#
NC4     025330          4994    5001#
NC5     025334          5003#
NC6     025362          4991    5013#
NC7     025406          5017    5021#
NC8     025452          5019    5031#
NC9     025476          5037#   5047    5049
ND    = 000052          5071#
NDDONE  026256          5126    5152    5186    5193#
ND1     025744          5110    5117#
ND10    026236          5177    5188#
ND2     025750          5119#   5160    5166
ND3     025776          5107    5129#
ND4     026034          5136    5143#
ND5     026040          5145#
ND6     026066          5133    5155#
ND7     026112          5159    5163#
ND8     026156          5161    5173#
ND9     026202          5179#   5189    5191
NMDONE  016046          3308    3327    3333    3348#
NMERR0  015646          3275    3310#
NM1     015626          3281    3297#
NM2     015630          3303#
NOCNC   032474          6247    6265#
OKSIZ   004146          1036    1070#
PARCNT  032340          5230    5351    6224#
PDMSG1  034100          6569#
PDMSG2  034256          6589#
PIRQ  = 177772            42#
PIRQVE= 000240           152#
POWERM  033373          5948    6490#
PR0   = 000000            74#
PR1   = 000040            75#
PR2   = 000100            76#
PR3   = 000140            77#
PR4   = 000200            78#
PR5   = 000240            79#
PR6   = 000300            80#
PR7   = 000340            81#
PS    = 177776            39#     40      906*    5987
```

```
PSW    = 177776            40#  6148*
PWRVEC= 000024            146#   918*   919*  5921*  5922*  5930*  5945*  5946*
RESMON  032362           1086   6241#  6264
RESREG= 104414           5904#  6079   6474
RESVEC= 000010            141#   958*
RSET   = 104416          2237   2323   2407   2494   2580   2668   2754   2853   2952   3051   3150   3249   3348
                         3447   3546   3674   3783   3895   4006   4117   4228   4339   4461   4519   4585   4650
                         4770   4910   5051   5193   5315   5436   5906#  6128   6243   6267   6287
SAVREG= 104412           5903#  6064   6440
SBT1    005172           1318#  1323
SBT1.2  005206           1323#
SDPAR0= 172260            273#
SDPAR1= 172262            274#
SDPAR2= 172264            275#
SDPAR3= 172266            276#
SDPAR4= 172270            277#
SDPAR5= 172272            278#
SDPAR6= 172274            279#
SDPAR7= 172276            280#
SDPDR0= 172220            251#
SDPDR1= 172222            252#
SDPDR2= 172224            253#
SDPDR3= 172226            254#
SDPDR4= 172230            255#
SDPDR5= 172232            256#
SDPDR6= 172234            257#
SDPDR7= 172236            258#
SIPAR0= 172240            262#
SIPAR1= 172242            263#
SIPAR2= 172244            264#
SIPAR3= 172246            265#
SIPAR4= 172250            266#
SIPAR5= 172252            267#
SIPAR6= 172254            268#
SIPAR7= 172256            269#
SIPDR0= 172200            240#
SIPDR1= 172202            241#
SIPDR2= 172204            242#
SIPDR3= 172206            243#
SIPDR4= 172210            244#
SIPDR5= 172212            245#
SIPDR6= 172214            246#
SIPDR7= 172216            247#
SIZE   = 104424          5909#
SIZEHI= 177762            171#  1050
SIZELO= 177760            169#  1035   1058   6302
SKAD    032100           1106*  1198*  1254*  1300*  1369*  1466*  1563*  1656*  1747*  1959*  2067*  2176*  2252*
                         2337*  2421*  2508*  2594*  2682*  2772*  2871*  2970*  3069*  3168*  3267*  3366*  3465*
                         3567*  3691*  3800*  3912*  4023*  4134*  4245*  4365*  4477*  4537*  4602*  4675*  4792*
                         4932*  5073*  5221*  5342*  6129  6131#  6268   6288
SKBADR  032172           5910   6164#
SKBCNR  032234           5912   6180#
SKBERR  032216           5911   6173#
SKBHMR  032270           5914   6194#
SKBMNR  032252           5913   6187#
SKIPT  = 104420          2798   2897   2996   3095   3194   3293   3392   3491   4558   4623   5907#  6101   6117
```

```
                    6201
SKPBAD= 104426      1259    5910#
SKPBCN= 104432      1204    1305    1374    1471    1568    1661    1752    1964    2071    2181    2257    2342    2426
                    2513    2687    2777    2876    2975    3074    3173    3272    3371    3470    3572    3696    3805
                    3917    4028    4139    4250    4371    4482    4542    4607    4680    4797    4937    5078    5912#
SKPBER= 104430      1260    2180    2256    2341    2425    2512    2686    2776    2875    2974    3073    3172    3271
                    3370    3469    3571    3695    3804    3916    4027    4138    4249    4370    4481    4541    4606
                    4679    4796    4936    5077    5911#
SKPBHM= 104436      1375    1472    1569    1662    1753    1965    2183    2259    2344    2428    2515    2689    2779
                    2878    2977    3076    3175    3274    3373    3472    3574    3698    3807    3919    4030    4141
                    4252    4373    4484    4544    4609    4682    4799    4939    5080    5914#
SKPBMN= 104434      1205    2072    2182    2258    2343    2427    2514    2688    2778    2877    2976    3075    3174
                    3273    3372    3471    3573    3697    3806    3918    4029    4140    4251    4372    4483    4543
                    4608    4681    4798    4938    5079    5913#
SKRNG   032304      6171    6178    6185    6192    6200#
SPUR    031754      1092    1109    1201    1566    1659    1750    1962    2141    2143    2157    4368    5224    5292
                    5303    5345    5413    5424    6104#   6110    6138
SR0     = 177572     189#   5993    6008*   6016*
SR1     = 177574     190#
SR2     = 177576     191#
SR3     = 172516     192#   6005*
STACK   = 001100      33#     34      35      36     911    6141
START   = 003014     448     905#    5950
STKLMT= 177774        41#
STOP    032452      6258    6260#
ST2     005226      1326    1329#
SUPSTK= 000700        35#
SWR     = 177570      43#     44    5502*   5516    5519    5527    5534    5573    5576    5583    5587    5593    6283
SW0     = 000001     109#
SW00    = 000001      99#    109
SW01    = 000002      98#    108
SW02    = 000004      97#    107
SW03    = 000010      96#    106
SW04    = 000020      95#    105
SW05    = 000040      94#    104
SW06    = 000100      93#    103
SW07    = 000200      92#    102
SW08    = 000400      91#    101
SW09    = 001000      90#    100
SW1     = 000002     108#
SW10    = 002000      89#
SW11    = 004000      88#
SW12    = 010000      87#
SW13    = 020000      86#
SW14    = 040000      85#
SW15    = 100000      84#
SW2     = 000004     107#
SW3     = 000010     106#
SW4     = 000020     105#
SW5     = 000040     104#
SW6     = 000100     103#
SW7     = 000200     102#   6283
SW8     = 000400     101#
SW9     = 001000     100#
SYSTID= 177764       173#
SOMOM1= 000034       431#   1405
```

```
SOM1   = 000030          429# 1503  1593  1675  1683  1710  1765  1867  1994  2783  2882  3179  3278
                        3944  4166  4545  4610
S1M0   = 000044          428# 1490  1582  1590  1617  1686  1767  1864  1976  2981  3080  3377  3476
                        4055  4277
S1MOM1= 000054           430# 1392
TAB    = 000011          427# 6498  6503  6510  6524  6527  6533  6543  6552  6561  6564  6690  6695
                        6757  6762  6768  6778  6784  6789  7647  7771  7779  7785  7791  7796  7815
                        7820  7828  7837  7844  7850  7856  7859  7882  7887  7896  7904  7913  7920
                        7930  7939  7945  7951  7955  7962  7968  7972  7986  7998
TBITVE= 000014           142#
TESTR1= 140000           434# 1574  1667  1762  1972
TESTR2= 142000           435# 1578  1671  1763  1974
TESTR3= 144000           436#
TKVEC  = 000060          149# 1076  1086* 1087* 6262* 6264*
TOP      004202         1074  1083#
TPVEC  = 000064          150#
TRAPVE= 000034           148#  916*  917*
TRTVEC= 000014           143#
TST1     004252         1102#
TST10    006564         1563  1652#
TST11    007114         1656  1743#
TST12    007724         1747  1932#
TST13    007756         1955#
TST14    010212         1959  2063#
TST15    010474         2067  2172#
TST16    010772         2176  2248#
TST17    011310         2252  2333#
TST2     004626         1106  1195#
TST20    011624         2337  2417#
TST21    012144         2421  2504#
TST22    012464         2508  2590#
TST23    013000         2594  2678#
TST24    013320         2682  2768#
TST25    013664         2772  2867#
TST26    014230         2871  2966#
TST27    014574         2970  3065#
TST3     004770         1198  1250#
TST30    015140         3069  3164#
TST31    015504         3168  3263#
TST32    016050         3267  3362#
TST33    016414         3366  3461#
TST34    016760         3465  3563#
TST35    017410         3567  3687#
TST36    020030         3691  3796#
TST37    020454         3800  3908#
TST4     005132         1254  1295#
TST40    021074         3912  4019#
TST41    021514         4023  4130#
TST42    022134         4134  4241#
TST43    022554         4245  4361#
TST44    023224         4365  4473#
TST45    023400         4477  4533#
TST46    023600         4537  4598#
TST47    024000         4602  4671#
TST5     005332         1300  1365#
TST50    024364         4675  4788#
```

```
TST51    025054            4792    4928#
TST52    025554            4932    5069#
TST53    026260            5073    5217#
TST54    026634            5221    5338#
TST55    027210            5342    5440#
TST6     005664            1369    1462#
TST7     006234            1466    1559#
TVADFL   033164            6387*   6399*   6426#   6444
TVADHI   033170            6433#   6443*   6466*
TVADLO   033166            6432#   6442*   6467*   6468
TYPDS  = 104410            5466    5902#   6364
TYPE   = 104400             930    1013    1018    1020    1024    1026    1037    1041    1045    1049    1054    1062    5464
                           5467    5578    5586    5690    5786    5863    5898#   5947    6169    6176    6183    6190    6197
                           6248    6260    6324    6341    6343    6347    6349    6373    6393    6408    6414    6472
TYPOC  = 104402            5899#   6333    6359
TYPON  = 104406            5901#
TYPOS  = 104404            1051    1059    1067    5900#   6380
TYPVAD   033172            6388    6400    6440#
T.END    003440             988     991#
T1       003272             962#   1008
T2       003334             969     972#
T3       003366             974     977     980#
T4       003420             983     986#
UA     = 000054            5219#
UAER1    026520            5235    5289#
UAER2    026560            5257    5300#
UATMP1   026514            5241    5253    5286#   5293
UATMP2   026516            5262    5274    5287#   5304
UA1      026322            5229#   5313
UA2      026346            5232    5235#
UA3      026402            5250#
UA4      026422            5257#   5258    5298
UA5      026466            5271#
UA6      026506            5278#
UA7      026614            5233    5278    5310#
UA8      026632            5311    5315#
UB     = 000055            5340#
UBER1    027074            5356    5410#
UBER2    027134            5378    5421#
UBTMP1   027070            5362    5374    5407#   5414
UBTMP2   027072            5383    5395    5408#   5425
UB1      026676            5350#   5434
UB2      026722            5353    5356#
UB3      026756            5371#
UB4      026776            5378#   5379    5419
UB5      027042            5392#
UB6      027062            5399#
UB7      027170            5354    5399    5431#
UB8      027206            5432    5436#
UDPAR0=  177660             229#
UDPAR1=  177662             230#
UDPAR2=  177664             231#
UDPAR3=  177666             232#
UDPAR4=  177670             233#
UDPAR5=  177672             234#
UDPAR6=  177674             235#
```

```
UDPAR7= 177676           236#
UDPDR0= 177620           207#
UDPDR1= 177622           208#
UDPDR2= 177624           209#
UDPDR3= 177626           210#
UDPDR4= 177630           211#
UDPDR5= 177632           212#
UDPDR6= 177634           213#
UDPDR7= 177636           214#
UIPAR0= 177640           218#
UIPAR1= 177642           219#
UIPAR2= 177644           220#
UIPAR3= 177646           221#
UIPAR4= 177650           222#
UIPAR5= 177652           223#
UIPAR6= 177654           224#
UIPAR7= 177656           225#
UIPDR0= 177600           196#
UIPDR1= 177602           197#
UIPDR2= 177604           198#
UIPDR3= 177606           199#
UIPDR4= 177610           200#
UIPDR5= 177612           201#
UIPDR6= 177614           202#
UIPDR7= 177616           203#
USESTK= 000600            36#
$BDADR  001122           498#
$BDDAT  001126           500#
$BELL   001300           554#  5578    5603
$CHARC  030326          5700*  5707    5716*  5721#
$CMTAG  001100           486#  906     907    914    920    921    922
$CM1  = 000024           512#  513#    514#   515#   516#   517#   518#   519#   520#   521#   522#   523#   524#
                         525#  526#    527#   528#   529#   530#   531#   532#
$CM2  = 000050           512#  513#    514#   515#   516#   517#   518#   519#   520#   521#   522#   523#   524#
                         525#  526#    527#   528#   529#   530#   531#   532#
$CM3  = 000024           510#  512
$CM4  = 000024           532#  533#    534#   535#   536#   537#   538#   539#   540#   541#   542#   543#   544#
                         545#  546#    547#   548#   549#   550#   551#   552#
$CORE   031444          5991  6019#
$CRLF   001305           556#  1049    5586   5603   5690   5724   6325   6344   6350
$CROUT  031474          6019  6026#
$DBLK   030774          5829  5863    5871#
$DB20   031606          6064#  6370    6406   6469
$DOAGN  027324          5460  5469    5473   5479#
$DTBL   030764          5832  5867#
$ENDAD  027314           472   928    5475#  5590
$ENDCT  027244           920  5462#
$ENDMG  027330          5464  5481#
$ENULL  027345          5467  5484#
$EOP    027210          5452#
$EOPCT  027236           920*  5459#  5463
$ERFLG  001103           489#  5492   5523   5525   5531*  5552   5570*  5603
$ERMAX  001115           495#   923*  5525   5547*  5552
$ERROR  027632           914   5569#
$ERRPC  001116           496#  5580*  5581*  5582   5603   6332   7670   8122   8125   8131   8144   8147   8150
                        8153  8168    8171   8174   8182   8196   8199   8204   8208   8211   8217   8220   8229
```

```
                        8233
$ERRTB  001314           581#  6338
$ERTTL  001112           493#  5579*  5603
$ESCAP  001276           553#   922*  5546*  5596  5598  5603
$FILLC  001150           508#  5693   5724
$FILLS  001147           507#  5724
$GDADR  001120           497#
$GDDAT  001124           499#
$GET42  027266          5468#
$HD   = 000003            10    11
$ICNT   001104           490#  5538*  5539  5541*  5551
$ILLUP  031212          5921   5952#
$ITEMB  001114           494#  5582*  5603  6125*  6330
$KTNEX  031436          5992   6018#
$KTOUT  031426          6009   6015#  6047
$KT11   031266          1031*  5990#  5994*  6018*
$LF     001306           557#  5603   5724
$LPADR  001106           491#   924*  5529*  5544*  5549  5551
$LPERR  001110           492#   925*  1118*  1206*  1262*  1378*  1391*  1404*  1475*  1489*  1502*  1756*  2085*
                        5229*  5258*  5350*  5379*  5529  5545*  5551  5595
$LSTAD  031602          6033*  6048#
$LSTBK  031604          1033*  1035   1066  4393  6034*  6049#
$MTMOU  031550          6010   6040#
$MXCNT  027630          5542   5551#
$NULL   001146           506#  5695   5724
$NWTST= 000001          1095#  1097   1187#  1189  1237#  1239  1286#  1288  1357#  1359  1449#  1451  1545#
                        1547   1638#  1640  1731#  1733  1915#  1917  1942#  1944  2048#  2050  2162#  2164
                        2240#  2242   2325#  2327  2409#  2411  2496#  2498  2582#  2584  2670#  2672  2757#
                        2759   2856#  2858  2955#  2957  3054#  3056  3153#  3155  3252#  3254  3351#  3353
                        3450#  3452   3552#  3554  3676#  3678  3785#  3787  3897#  3899  4008#  4010  4119#
                        4121   4230#  4232  4341#  4343  4463#  4465  4523#  4525  4588#  4590  4656#  4658
                        4773#  4775   4913#  4915  5054#  5056  5196#  5198  5317#  5319
$OCNT   030554          5758*  5787*  5800#
$OCTVL  031710          6066   6091#
$OMODE  030556          5753*  5757*  5762  5765*  5776*  5802#
$OVER   027614          5503   5522   5530  5540  5548#
$PASS   001100           487#  5456*  5457*  5465  5481  5536  5552
$PWRAD  031206          5950#
$PWRDN  031064           918   5921#  5945
$PWRMG  031202          5948#
$PWRUP  031132          5930   5935#
$QUES   001304           555#  5603
$RDCHR= ****** U        5903
$RDDEC= ****** U        5903
$RDLIN= ****** U        5903
$RDOCT= ****** U        5903
$REGAD  001152           510#
$REG0   001154           512#
$REG1   001156           513#
$REG10  001174           520#
$REG11  001176           521#
$REG12  001200           522#
$REG13  001202           523#
$REG14  001204           524#
$REG15  001206           525#
$REG16  001210           526#
```

```
$REG17  001212          527#
$REG2   001160          514#
$REG20  001214          528#
$REG21  001216          529#
$REG22  001220          530#
$REG23  001222          531#
$REG3   001162          515#
$REG4   001164          516#
$REG5   001166          517#
$REG6   001170          518#
$REG7   001172          519#
$RESRE  030060         5638#   5904
$SAVRE  030022         5622#   5903
$SAVR6  031216         5929*   5935   5936*   5937*   5954#
$SCOPE  027350          912    5501#
$SETUP= 000037          421#    912    914     916     918     920     921     922     924     928    5454    5587
$SIZE   031220         1032    5978#
$SIZEX  031500         6017    6027#
$STUP = 177777          421#
$SVLAD  027566         5511    5543#
$SVPC = 000204          470#    475
$SWR  = 167400           10     11#    12#     20      21      22      23      24      25      26     552     553     554
                        921     922    924     925    1103    1196    1251    1296    1366    1463    1560    1653    1744
                       1933    1956   2064    2173    2249    2334    2418    2505    2591    2679    2769    2868    2967
                       3066    3165   3264    3363    3462    3564    3688    3797    3909    4020    4131    4242    4362
                       4474    4534   4599    4672    4789    4929    5070    5218    5339    5449    5455    5470    5480
                       5481    5493   5494    5495    5496    5497    5502    5514    5516    5517    5523    5524    5525
                       5532    5533   5534    5545    5548    5551    5560    5561    5562    5563    5564    5565    5573
                       5576    5583   5587    5593    5603
$SWRMK= 000200           13#     26     27    5497    5498    5518    5520
$TAB    033440         6415    6498#
$TIMES  001274          552#    921*  1103*   1251*   1296*   1366*   1463*   1560*   1653*   1744*   1956*   2064*   2173*
                       2249*   2334*  2418*   2505*   2591*   2679*   2769*   2868*   2967*   3066*   3165*   3264*   3363*
                       3462*   3564*  3688*   3797*   3909*   4020*   4131*   4242*   4362*   4474*   4534*   4599*   4672*
                       4789*   4929*  5070*   5218*   5339*   5455*   5532*   5539    5542*   5551
$TKB    001140          503#   1088*  6242    6265*
$TKS    001136          502#    909   1089*   6266*
$TMP0   001224          532#   1108*  1200*   1256*   1302*   1371*   1468*   1565*   1658*   1749*   1961*   2069*   2178*
                       2254*   2339*  2423*   2510*   2596*   2684*   2774*   2873*   2972*   3071*   3170*   3269*   3368*
                       3467*   3569*  3693*   3802*   3914*   4025*   4136*   4247*   4367*   4479*   4539*   4604*   4677*
                       4794*   4934*  5075*   5223*   5344*   7670    8122    8125    8128    8131    8144    8147    8150
                       8153    8168   8171    8174    8182    8196    8199    8204    8208    8211    8217    8220    8229
                       8233    8238
$TMP1   001226          533#   1140*  1272*   1988*   2004*   2146*   2795*   2894*   2993*   3092*   3191*   3290*   3389*
                       3488*   4555*  4620*   6096*   6112*   6124*   8122    8125    8128    8131    8204    8238
$TMP10  001244          540#   1575*  1668*   8168    8174
$TMP11  001246          541#   1576*  1669*
$TMP12  001250          542#   1579*  1672*   8171    8177
$TMP13  001252          543#   1580*  1673*
$TMP14  001254          544#
$TMP15  001256          545#
$TMP16  001260          546#
$TMP17  001262          547#
$TMP2   001230          534#   1223*  1231*   1273*   1349*   1422*   1427*   1432*   1520*   1525*   1530*   1589*   1613*
                       1624*   1682*  1706*   1717*   1907*   1987*   2003*   2035*   2116*   2125*   2147*   2197*   2211*
                       2225*   2279*  2296*   2311*   2363*   2380*   2395*   2450*   2467*   2482*   2536*   2553*   2568*
```

```
                    2624*  2641*  2656*  2710*  2727*  2742*  2794*  2809*  2826*  2841*  2893*  2908*  2925*
                    2940*  2992*  3007*  3024*  3039*  3091*  3106*  3123*  3138*  3190*  3205*  3222*  3237*
                    3289*  3304*  3321*  3336*  3388*  3403*  3420*  3435*  3487*  3502*  3519*  3534*  3627*
                    3642*  3662*  3739*  3756*  3771*  3851*  3868*  3883*  3962*  3979*  3994*  4073*  4090*
                    4105*  4184*  4201*  4216*  4295*  4312*  4327*  4414*  4429*  4449*  4511*  4554*  4576*
                    4619*  4641*  4702*  4726*  4741*  4757*  4840*  4866*  4881*  4897*  4982*  5007*  5022*
                    5038*  5123*  5149*  5164*  5180*  5252*  5273*  5373*  5394*  6097*  6113*  7670   8122
                    8144   8147   8150   8153   8168   8171   8174   8182   8196   8199   8208   8211   8217
                    8220   8229   8233   8238
$TMP20  001264       548#
$TMP21  001266       549#
$TMP22  001270       550#
$TMP23  001272       551#
$TMP3   001232       535#   1142*  1224*  1274*  1350*  1590*  1602*  1614*  1625*  1683*  1695*  1707*  1718*
                    2036*  2117*  2126*  2148*  2212*  2227*  2297*  2313*  2381*  2397*  2468*  2484*  2554*
                    2570*  2642*  2658*  2728*  2744*  2827*  2843*  2926*  2942*  3025*  3041*  3124*  3140*
                    3223*  3239*  3322*  3338*  3421*  3437*  3520*  3536*  3644*  3663*  3757*  3773*  3869*
                    3885*  3980*  3996*  4091*  4107*  4202*  4218*  4313*  4329*  4415*  4431*  4450*  4513*
                    4578*  4643*  4742*  4758*  4882*  4898*  5023*  5039*  5165*  5181*  5253*  5274*  5374*
                    5395*  6098*  6114*  7670   8125   8131   8144   8150   8153   8168   8171   8177   8199
                    8204   8211   8220   8229   8233   8238
$TMP4   001234       536#   1143*  1351*  2037*  2145*  2213*  2228*  2298*  2314*  2382*  2398*  2469*  2485*
                    2555*  2571*  2643*  2659*  2729*  2745*  2828*  2844*  2927*  2943*  3026*  3042*  3125*
                    3141*  3224*  3240*  3323*  3339*  3422*  3438*  3521*  3537*  3645*  3664*  3758*  3774*
                    3870*  3886*  3981*  3997*  4092*  4108*  4203*  4219*  4314*  4330*  4432*  4451*  4514*
                    4579*  4644*  4743*  4759*  4883*  4899*  5024*  5040*  5166*  5182*  5254*  5275*  5375*
                    5396*  6111*  8125   8153   8196   8204   8217   8229   8235
$TMP5   001236       537#   2038*  2229*  2315*  2399*  2486*  2572*  2660*  2746*  2845*  2944*  3043*  3142*
                    3241*  3340*  3439*  3538*  3646*  3775*  3887*  3998*  4109*  4220*  4331*  4433*  4515*
                    4580*  4645*  4744*  4884*  5025*  5167*  8211   8220
$TMP6   001240       538#   2230*  2316*  2400*  2487*  2573*  2661*  2747*  2846*  2945*  3044*  3143*  3242*
                    3341*  3440*  3539*  3647*  3776*  3888*  3999*  4110*  4221*  4332*  4434*  4516*  4581*
                    4646*  4745*  4885*  5026*  5168*  8204   8222   8229   8235
$TMP7   001242       539#   2231*  2317*  2401*  2488*  2574*  2662*  2748*  2847*  2946*  3045*  3144*  3243*
                    3342*  3441*  3540*  3649*  3777*  3889*  4000*  4111*  4222*  4333*  4435*  4746*  4886*
                    5027*  5169*  8211   8222
$TN     = 000055      10#    1095   1103#  1104   1105   1187   1196#  1197   1237   1251#  1252   1253   1286
                    1296#  1298   1299   1357   1366#  1367   1368   1449   1463#  1464   1465   1545   1560#
                    1561   1562   1638   1653#  1654   1655   1731   1744#  1745   1746   1915   1933#  1942
                    1956#  1957   1958   2048   2064#  2065   2066   2162   2173#  2174   2175   2240   2249#
                    2250   2251   2325   2334#  2335   2336   2409   2418#  2419   2420   2496   2505#  2506
                    2507   2582   2591#  2592   2593   2670   2679#  2680   2681   2757   2769#  2770   2771
                    2856   2868#  2869   2870   2955   2967#  2968   2969   3054   3066#  3067   3068   3153
                    3165#  3166   3167   3252   3264#  3265   3266   3351   3363#  3364   3365   3450   3462#
                    3463   3464   3552   3564#  3565   3566   3676   3688#  3689   3690   3785   3797#  3798
                    3799   3897   3909#  3910   3911   4008   4020#  4021   4022   4119   4131#  4132   4133
                    4230   4242#  4243   4244   4341   4362#  4363   4364   4463   4474#  4475   4476   4523
                    4534#  4535   4536   4588   4599#  4600   4601   4656   4672#  4673   4674   4773   4789#
                    4790   4791   4913   4929#  4930   4931   5054   5070#  5071   5072   5196   5218#  5219
                    5220   5317   5339#  5340   5341
$TPB    001144       505#   5713*  5724
$TPFLG  001151       509#   5673   5724
$TPS    001142       504#   5711   5724
$TRAP   031004       916    5882#
$TRP    = 000040    5889#  5899#  5900#  5901#  5902#  5903#  5904#  5905#  5906   5907#  5908#  5909#  5910#
                    5911#  5912#  5913#  5914#  5915#
```

```
$TRPAD  031024          5886    5897#
$TSTNM  001102           488#    905*  1108  1200  1256  1302  1371  1468  1565  1658  1749  1961  2069
                        2178    2254  2339  2423  2510  2596  2684  2774  2873  2972  3071  3170  3269
                        3368    3467  3569  3693  3802  3914  4025  4136  4247  4367  4479  4539  4604
                        4677    4794  4934  5075  5223  5344  5454* 5492  5521  5543* 5548  5552  5572
                        5603
$TYPBN= ****** U        5903
$TYPDS  030560          5817#   5902
$TYPE   030116          5673#   5889  5898
$TYPEC  030262          5692    5699  5706  5711#  5712
$TYPEX  030330          5717    5719  5722#
$TYPOC  030356          5756#   5899
$TYPON  030372          5755    5758#  5901
$TYPOS  030332          5751#   5900
$XTSTR  027356          5505#
$$GET4= 000001          5470#   5472#
$$TRP = 000002          5888#   5899  5900  5901  5902  5903  5904  5905  5907  5908  5909  5910  5911
                        5912    5913  5914  5915
$OFILL  030555          5752*   5756* 5766  5801#
.       = 060702         440#    444   446#  470   471#  473#  475#  484#  558   910   924   925  1040#
                        1044#   1057# 1065# 1795  1798# 2101  2266  2269# 2351  2354# 2435  2438# 2522
                        2525#   2610  2613# 2696  2699# 3728  3731# 3837  3840# 3949  3952# 4060  4063#
                        4171    4174# 4282  4285# 4492  4495# 4691  4694# 4715  4718# 4829  4832# 4855
                        4858#   4971  4974# 4996  4999# 5112  5115# 5138  5141# 5281  5284# 5402  5405#
                        5481    5485  5551  5552  5603  5724  5871# 5932  5953  6091# 8118# 8242#
```

```
ADDDPA      1#
ADJUST      1#  1794  2265  2350  2434  2521  2609  2695  3727  3836  3948  4059  4170  4281  4491
         4690   4714  4828  4854  4970  4995  5111  5137  5280  5401
BYTLT1      1#
BYTLT2      1#
CHAIN       1#  5470
CLRMAC      1#  6136
CLRRFL      1#  1110
CMPDPA      1#
CNVDPA      1#
CNVMAP      1#
COMMEN      1#   419#
ENDCOM      1#   419#
ERCLR       1#  5600
ERROR      37#  1147  1153  1159  1165  1171  1177  1225  1232  1275  1352  1423  1428  1433  1521
         1526   1531  1591  1603  1615  1626  1684  1696  1708  1719  1908  1989  2005  2040  2118
         2127   2151  2198  2214  2232  2281  2300  2319  2365  2384  2403  2452  2471  2490  2538
         2557   2576  2626  2645  2664  2712  2731  2750  2796  2811  2830  2849  2895  2910  2929
         2948   2994  3009  3028  3047  3093  3108  3127  3146  3192  3207  3226  3245  3291  3306
         3325   3344  3390  3405  3424  3443  3489  3504  3523  3542  3628  3635  3649  3665  3741
         3760   3779  3853  3872  3891  3964  3983  4002  4075  4094  4113  4186  4205  4224  4297
         4316   4335  4416  4436  4452  4517  4556  4583  4621  4648  4703  4727  4748  4761  4841
         4867   4888  4901  4983  5008  5029  5042  5124  5150  5171  5184  5255  5276  5376  5397
         6100   6116
ESCAPE      1#   419#
EXTRA       1#   558
IIMAC1      1#
ITEMAC      1#   586
JJM1        1#
KKM1        1#
KKM2        1#
KMAC1       1#  1545  1638
KMAC5       1#  4536  4601
MMAC1       1#  2284  2368  2455  2541  2629  2715  2814  2913  3012  3111  3210  3309  3408  3507
         3744   3856  3967  4078  4189  4300
MMAC2       1#  2277  2361  2448  2534  2622  2708  2807  2906  3005  3104  3203  3302  3401  3500
         3737   3849  3960  4071  4182  4293
MMAC3       1#  2771  2870  2969  3068  3167  3266  3365  3464
MMAC4       1#  3911  4022  4133  4244
MMAC5       1#  3799
MMM1        1#
MMM2        1#
MSG      1094#  1097  1187#  1189  1285#  1288  1356#  1359  1449#  1451  1545#  1547  1638#  1640  1731#
         1733   1942#  1944  2048#  2050  2161#  2164  2240#  2242  2325#  2327  2409#  2411  2496#  2498
         2582#  2584  2670#  2672  2757#  2759  2856#  2858  2955#  2957  3054#  3056  3153#  3155  3252#
         3254   3351#  3353  3450#  3452  3552#  3554  3676#  3678  3785#  3787  3897#  3899  4008#  4010
         4119#  4121  4230#  4232  4341#  4343  4463#  4465  4523#  4525  4588#  4590  4656#  4658  4773#
         4775   4913#  4915  5054#  5056  5196#  5198  5317#  5319
MSG101   4654#  4666  4668  4783  4925
MSG102   4655#  4785  4923  5064  5066
MSG20    1544#  1547  1640
MSG200   5196#  5198  5319
MSG25    2159#  2242  2327  2411  2498  2584  2672
MSG27    2756#  2759  2858  2957  3056  3155  3254  3353  3452
MSG30    1915#  1917
MSG300   7479#  7521  7563  7605
```

```
MSG31   3549#  3554  3678  3787  3899  4010  4121  4232
MSG32   3550#  3678  3787
MSG33   3551#  3899  4010  4121  4232
MSG6    1237#  1239
MSG98   4521#  4525  4590
MSG99   4653#  4658  4775  4915  5056
MULT       1#   419#
NEWTST     1#   419#  1095  1187  1237  1286  1357  1449  1545  1638  1731  1915  1942  2048  2162
        2240  2325  2409  2496  2582  2670  2757  2856  2955  3054  3153  3252  3351  3450  3552
        3676  3785  3897  4008  4119  4230  4341  4463  4523  4588  4656  4773  4913  5054  5196
        5317
NMAC1      1#  4685  4708  4823  4990
NMAC2      1#  4846  4963  5104  5129
NMAC3      1#  4732  4872  5013  5155
POP        1#   419#  5643  5858  5939
PUSH       1#   419#  5623  5817  5923
QQM1       1#
SCOPE     38#  1102  1195  1250  1295  1365  1462  1559  1652  1743  1932  1955  2063  2172  2248
        2333  2417  2504  2590  2678  2768  2867  2966  3065  3164  3263  3362  3461  3563  3687
        3796  3908  4019  4130  4241  4361  4473  4533  4598  4671  4788  4928  5069  5217  5338
        5453
SETMAP     1#  3586  3701  3809  3921  4032  4143  4254  4801  4941  5082
SETMM      1#  4375
SETTRA  5889#  5899  5900  5901  5902  5903  5904  5906  5907  5908  5909  5910  5911  5912  5913
        5914
SETUP      1#   419#   906
SKBREX     1#  1203  1258  1304  1373  1470  1567  1660  1751  1963  2070  2179  2255  2340  2424
        2511  2685  2775  2874  2973  3072  3171  3270  3369  3468  3570  3694  3803  3915  4026
        4137  4248  4369  4480  4540  4605  4678  4795  4935  5076  5225  5346
SKIP       1#   419#
SLASH      1#   419#
SPACE   419#
SSKAD      1#  1105  1197  1253  1299  1368  1465  1562  1655  1746  1958  2066  2175  2251  2336
        2420  2507  2593  2681  2771  2870  2969  3068  3167  3266  3365  3464  3566  3690  3799
        3911  4022  4133  4244  4364  4476  4536  4601  4674  4791  4931  5072  5220  5341
STARS      1#   419#   450   477   565  1028  1072  1095  1101  1187  1194  1237  1249  1286  1294
        1307  1314  1357  1364  1449  1461  1545  1558  1638  1651  1731  1742  1915  1931  1942
        1954  2048  2062  2162  2171  2240  2247  2325  2332  2409  2416  2496  2503  2582  2589
        2670  2677  2757  2767  2856  2866  2955  2965  3054  3064  3153  3163  3252  3262  3351
        3361  3450  3460  3552  3562  3676  3686  3785  3795  3897  3907  4008  4018  4119  4129
        4230  4240  4341  4360  4463  4472  4523  4532  4588  4597  4656  4670  4773  4787  4913
        4927  5054  5068  5196  5216  5317  5337  5442  5486  5553  5604  5651  5725  5804  5873
        5916  5955  6052
THIT       1#  1982  1998  2789  2888  2987  3086  3185  3284  3383  3482  4549  4614
TOMAPO     1#
TOPAR3     1#
TRMTRP  5889#
TSTHIT     1#  1980  1996
TYPBIN     1#   419#
TYPDEC     1#   419#  5465
TYPNAM     1#   419#   926
TYPNUM     1#   419#
TYPOCS     1#   419#  1050  1058  1066
TYPOCT     1#   419#
TYPTXT     1#   419#  1037  1041  1045  1054  1062
UMAC1      1#  5196  5317
```

```
UMAC2       1#    5289    5300    5410    5421
UMAC3       1#    5250    5271    5371    5392
$$CMRE    477#     512     513     514     515     516     517     518     519     520     521     522     523     524     525
          526      527     528     529     530     531
$$CMTM    477#     532     533     534     535     536     537     538     539     540     541     542     543     544     545
          546      547     548     549     550     551
$$ESCA      1#    419#
$$NEWT      1#    419#    1095    1187    1237    1286    1357    1449    1545    1638    1731    1915    1942    2048    2162
         2240    2325    2409    2496    2582    2670    2757    2856    2955    3054    3153    3252    3351    3450    3552
         3676    3785    3897    4008    4119    4230    4341    4463    4523    4588    4656    4773    4913    5054    5196
         5317
$$SET     5889#   5899    5900    5901    5902    5903    5904    5906    5907    5908    5909    5910    5911    5912    5913
         5914
$$SKIP      1#    419#
.EQUAT      1#
.HEADE      1#
.KT11       1#
.SETUP      1#     421
.SWRHI      1#      15
.SWRLO     27#
.$ACT1      1#     450
.$CATC      1#     437
.$CMTA      1#     477
.$DB2D      1#
.$DB2O      1#    6052
.$DIV       1#
.$EOP       1#    5442
.$ERRO      1#    5553
.$ERRT      1#
.$MULT      1#
.$POWE      1#    5916
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$SAVE      1#    5604
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#    5486
.$SIZE      1#    5955
.$SUPR      1#
.$TRAP      1#    5873
.$TYPB      1#
.$TYPD      1#    5804
.$TYPE      1#    5651
.$TYPO      1#    5725
.1170       1#      29

. ABS.  060702     000

 ERRORS DETECTED:  0

 CEKBCD.BIN,CEKBCD.LST/CRF/SOL/NL:TOC=CEKBCD.SML,CEKBCD.P11
 RUN-TIME: 60 86 10 SECONDS
```

RUN-TIME RATIO: 507/156=3.2
CORE USED: 35K (69 PAGES)