11/70–74

11/70 – 74 CACHE # 2

CEKBDD0

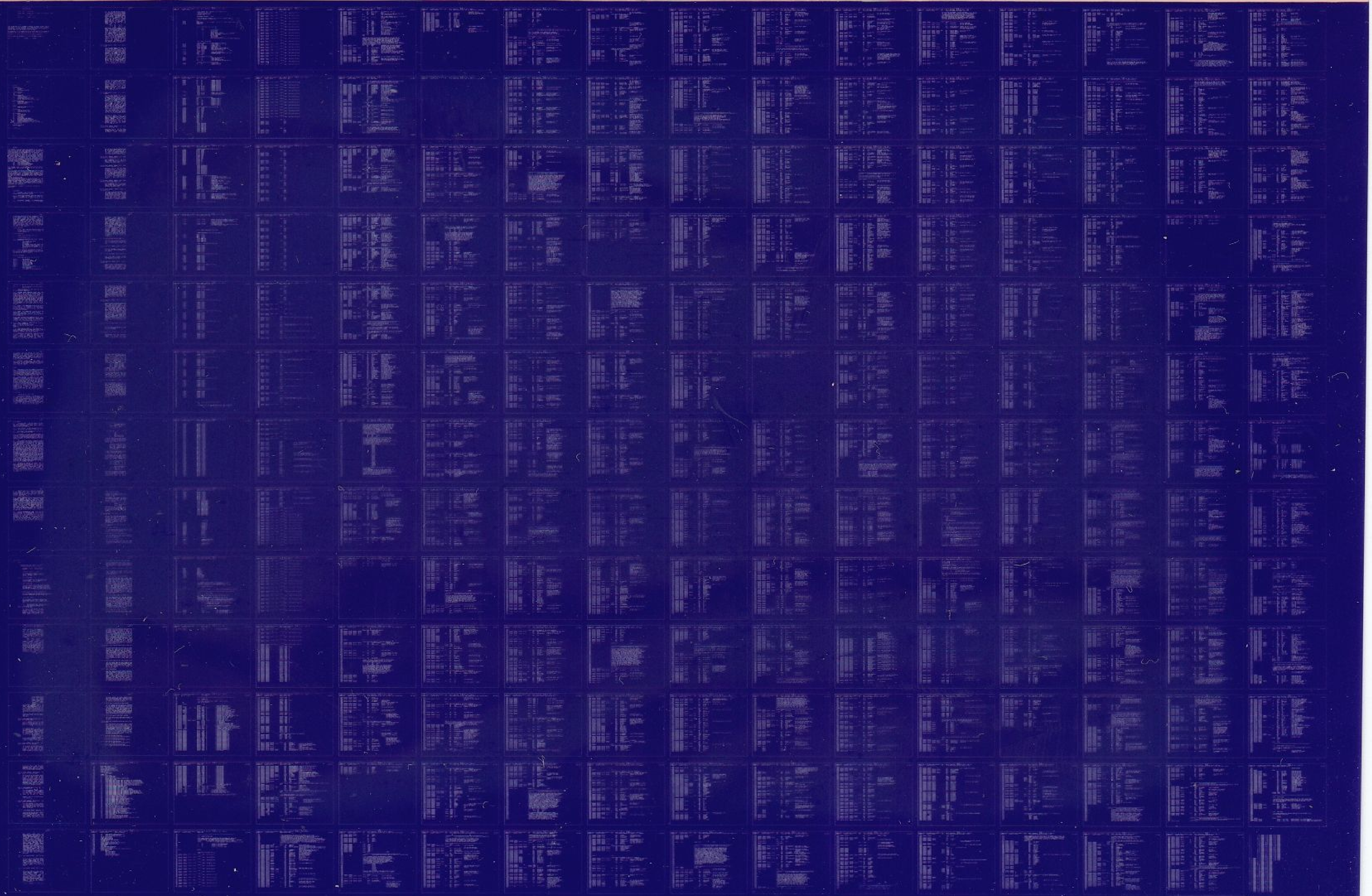
AH-7972D-MC

JAN 1980

COPYRIGHT 75–80

FICHE 1 OF 2

MADE IN USA



11/70–74

11/70 – 74 CACHE # 2

CEKBDD0

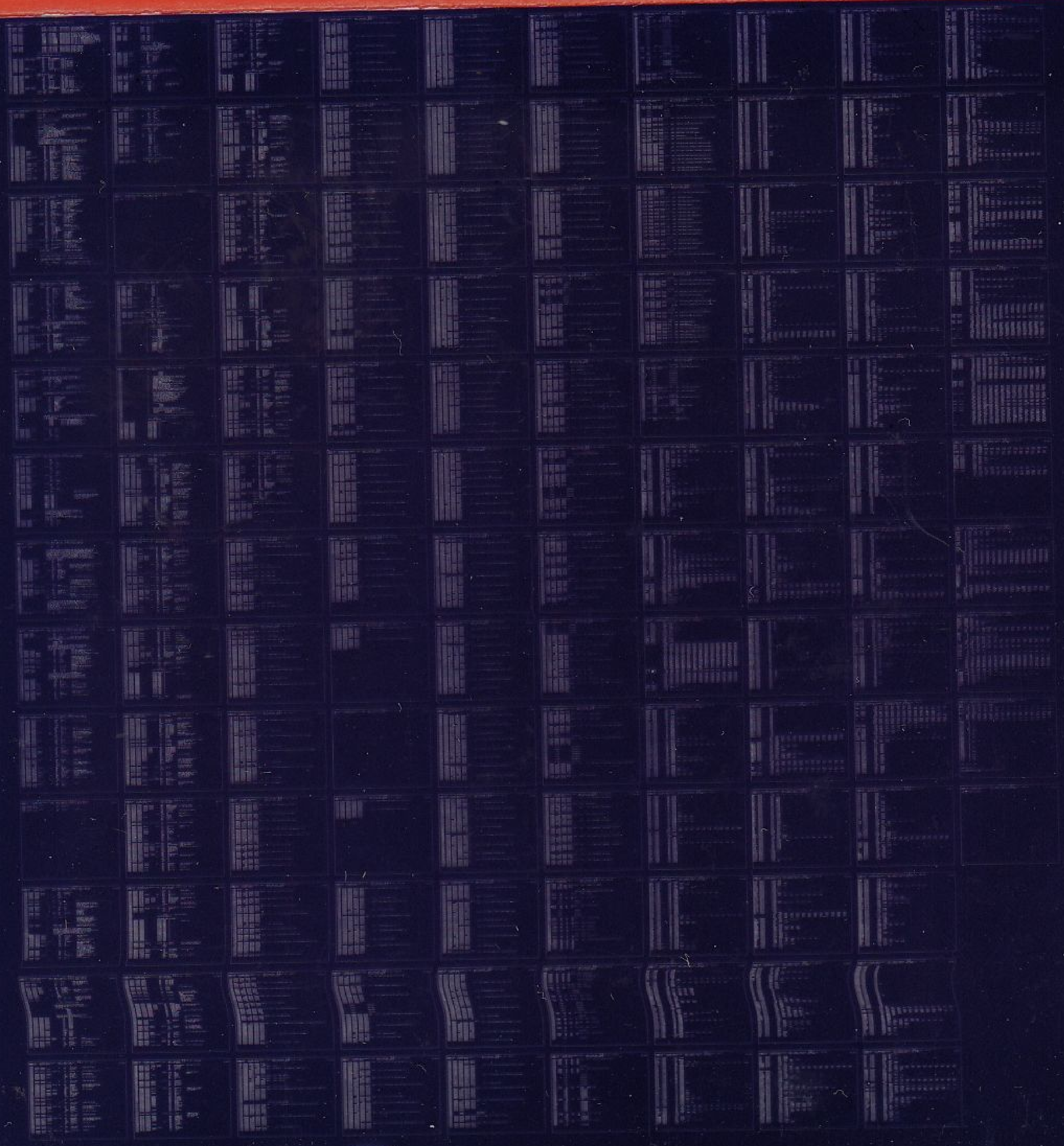
AH-7972D-MC

FICHE 2 OF 2

JAN 1980

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: AC-7971D-MC

PRODUCT NAME: CEKBDD0 PDP-11/70-74MP CACHE DIAGNOSTIC (PART 2)

DATE: MAY, 1979

MAINTAINER: DIAGNOSTIC GROUP

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC
PDP
DECUS
UNIBUS
DECTAPE
MASSBUS
DECX/11

CONTENTS

1. ABSTRACT

THE PROGRAMS, CEKBC AND CEKBD, ARE INTENDED TO BE USED AS AIDS FOR THE REPAIR AND MAINTENANCE OF THE CACHE MEMORY SYSTEM IN THE PDP 11/70-74MP COMPUTING SYSTEM. THE AIM IS TO DETECT AND REPORT FAILING COMPONENTS OF THE CACHE UNIT. THE FAILURES ARE TYPICALLY IDENTIFIED WITH A FAILING CIRCUIT WHEN THE REPORT IS MADE, BUT THE OVERALL DIAGNOSTIC PHILOSOPHY HAS BEEN TO LOCATE THE FAILING MODULE (HEX BOARD) OF WHICH THERE ARE FOUR (4) IN THE CACHE UNIT. NOTE THAT WHEN A FAILURE IS REPORTED AND THE ASSOCIATED CIRCUIT IDENTIFIED, THAT CIRCUIT SHOULD NOT BE TAKEN IN BLIND FAITH AS THE DEFECTIVE COMPONENT; THE IDENTIFIED COMPONENT SHOULD RATHER BE TAKEN AS THE PROBABLE CAUSE OF THE FAILURE. THERE ARE FOUR (4) MODULES (HEX BOARDS) IN THE CACHE UNIT:

CCB	CACHE CONTROL BOARD
CDP	CACHE DATA PATHS BOARD
ADM	CACHE ADDRESS MEMORY BOARD
DTM	CACHE DATA MEMORY BOARD

THE PROGRAM CEKBC IS DESIGNED TO TEST THE FIRST TWO OF THESE BOARDS, WHILE CEKBD IS DESIGNED TO TEST THE LAST TWO BOARDS.

NOTE THAT THOUGH THE TESTING HAS BEEN DIVIDED INTO TWO STAND ALONE PROGRAMS, EACH ASSOCIATED WITH TWO MODULES, IT SHOULD NOT BE ASSUMED THAT A PARTICULAR MODULE IS WORKING AFTER HAVING RUN ONLY ONE OF THE PROGRAMS! BOTH PROGRAMS SHOULD BE RUN! FOR EXAMPLE, JUST RUNNING CEKBC WITHOUT ERROR DOES NOT RULE OUT A 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 11/70-74MP CPU WITH OPERATORS CONSOLE LA30 OR EQUIVALENT TERMINAL.

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

2.3 PRELIMINARY PROGRAMS - THIS PROGRAM ASSUMES THAT THE CPU IS FUNCTIONAL! THIS COULD IN SOME

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

3. LOADING PROCEDURE

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

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS (SEE 5.1)

4.2 STARTING ADDRESS - 200

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

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

4.4 SPECIAL OPERATOR INTERVENTION OPTIONS - IF SWITCH 12 OF THE SWITCH REGISTER IS ON, THEN CEKBD WILL REQUIRE THE OPERATOR TO POWER THE MACHINE FIRST DOWN AND THEN UP (SEE 5.1 AND 8.7).

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS FOR CEKBC:

SW<15>=1	HALT ON ERROR
SW<14>=1	LOOP ON TEST
SW<13>=1	INHIBIT ERROR 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

9.1 CEKBD

PROGRAM BY ANTHONY S. VEZZA

.

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC PACKAGE (MAINDEC-11-DZQAC-A5-1).

TEST 1 PARITY ERROR ABORT

THIS TEST ENSURES THAT A CACHE PARITY ERROR FLAG CAUSES AN ABORT. THIS IS DONE BY FORCING A PARITY ERROR ON AN EVEN WORD.

TEST 2 PARITY ERROR TRAP

THIS TEST ENSURES THAT A PARITY TRAP FUNCTIONS PROPERLY. THIS IS DONE BY MAKING THE ODD WORD HAVE BAD PARITY. IF THE TRAP DOES'T OCCUR THEN THE PROBLEM IS ON TMCA. IF A TRAP OCCURS TO THE WRONG VECTOR THE PROBLEM COULD BE ON TMCA OR UBCB.

TEST 3 MEM MGT AND PE TRAP PRIORITY ARBITRATION

THIS TEST ENSURES THAT THE ARBITRATION LOGIC WORKS FOR MEMORY MANAGEMENT AND PARITY ERROR TRAPS.

TEST 4 UNIBUS PARITY ERROR

THIS TEST MAKES A REFERENCE TO MEMORY THRU MAPPING BOX THAT WILL CAUSE A PARITY ERROR. IF ABORT DOESN'T HAPPEN THEN THE PROBLEM IS ON UBCB.

NOTE: MAP REGISTER 0 AND 1 ARE NOT USED INCASE THE PROGRAM IS RUNNING UNDER ACT11.

TEST 5 CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FLOATING ONES

THIS TEST IS A TEST OF BOTH THE AMX, CPU INPUTS, AND THE CACHE ERROR ADDRESS REGISTER. A SET OF ADDRESSES IS GENERATED AND A MAIN MEMORY ADDRESS AND CONTROL LINE PARITY ERROR IS FORCED AT EACH,

```
THEREBY LOCKING UP  THE  ADDRESS  ON
THE  OUTPUT  OF THE AMX IN THE ERROR


ADDRESS  REGISTER.   THE  MANNER  IN
WHICH  THIS  IS  DONE IS AS FOLLOWS:
FIRST  THE  ADDRESS  IS   GENERATED;
THEN, IF IT IS A VALID ADDRESS (THAT
IS, IF IT IS NOT BEYOND  THE  LIMITS
OF MEMORY AS DISPLAYED IN THE SYSTEM
SIZE   REGISTER),   THESE     THREE
INSTRUCTIONS  ARE MOVED TO THAT AREA
OF MEMORY:
        ONE:     MOV     R1,(R2)
        2$:      CLR     (R2)
        3$:      RTS     PC 2$ IS THE
ADDRESS     BEING     TESTED.     THE
INSTRUCTION AT ONE IS GIVEN  CONTROL
BY  A  'JSR  PC'.   R1  IS  MADE  TO
CONTAIN  #2  AND  R2  CONTAINES  THE
ADDRESS OF THE MAINTENANCE REGISTER,
SO THAT AFTER THE 'MOV  R1,(R2)'  IS
EXECUTED A PARITY ERROR SHOULD OCCUR
ON  THE  MAIN  MEMORY  ADDRESS   AND
CONTROL    LINES   WHEN   THE   NEXT
INSTRUCTION    IS    FETCHED.    THE
ADDRESSES    USED    ARE   GENERATED
FOLLOWING THIS PATTERN
                200000 200002 200004
```

```
                        200010 200020 200040
                        200100 200200 200400
                        ETC.    TO:   240000
                        300000 400000 400002
                        400004  400010  ETC.
                        TO:   500000  600000
                        1000000      1000002
                        1000004 ETC.
```

THE PATTERN CONINUES UNTIL AN ADDRESS IS GENERATED THAT IS TOO LARGE. MEMORY MANAGEMENT IS SET UP TO FULL 22-BIT MODE, SO IF THE USER WANTS TO HAVE THE EXECUTION OF THIS TEST DELETED HE CAN SIMPLY BY TURNING ON THE APPORPRIATE CONSOLE SWITCH WHICH HAS BEEN DESIGNATED FOR THE

PURPOSE OF DELETING THE EXECUTION OF TESTS WHICH MAKE USER OF MEMORY MANAGEMENT.

TEST 6 CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FLOATING ZEROES

THIS IS ANOTHER TEST OF THE AMX WHICH IS CARRIED OUT USING THE SAME METHOD AS IN THE PREVIOUS TEST ALL THAT IS DIFFERENT IS THE SERIES OF

TEST ADDRESSES WHICH IS USED. IN THE PREVIOUS TEST A ONE WAS FLOATED THROUGH A FIELD OF ZEROES TO PRODUCE THE TEST ADDRESSES, HERE A ZERO WIL BE FLOATED THROUGH A FIELD OF ONES TO PRODUCE THE ADDRESSES BASE ADDRESSES WHICH ARE USE ARE:

```
                        177776 377776 777776
                        1777776      3777776
                        7777776 17777776
```

EACH OF THESE PATTERNS IS TAKEN AND A ZERO IS FLOATED THROUGHT THE FIELD OF ONES TO PRODUCE A TEST ADDRESS.

TEST 7 CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ONES

THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX. THIS TEST IS IDENTICAL TO TST1 IN EVERY THING IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM THE

CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE AS UNIBUS REFERENCES.

TEST 10 CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ZEROES

THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX. THIS TEST IS IDENTICAL TO TST2 IN EVERY THING IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM THE CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE AS UNIBUS REFERENCES.

TEST 11 CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS DUAL ADDRESS TEST

THIS TEST PERFORMS A DUAL ADDRESS TEST ON MEMORY LOCATED AT ADDRESSES LESS THAN 160000 (OCT.) OR WITHIN THE FIRST 28K. THE PURPOSE IS TO VARIFY THE THE AMX IS WORKING

PROPERLY FOR THE LOW ORDER ADDRESS LINES INVOLVED.

TEST 12 CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS DUAL ADDRESS TEST

THIS TEST PERFORMS A DUAL ADDRESS TEST IDENTICAL TO TST5, EXCEPT THAT IT IS DONE THROUGH THE MAPPING BOX HERE THEREBY TESTING THE UNIBUS INPUTS TO THE AMX.

TEST 13 CACHE ADDRESS MEMORY COMPARATOR TEST

THIS IS A TEST OF THE CACHE ADDRESS MEMORY ADDRESS COMPARATORS. THIS IS A CIRCUIT MADE UP OF SIX 74585 CHIPS, THREE FOR EACH GROUP. EACH CHIP COMPARES FOUR BITS OF THE ADDRESS ON THE ADDRESS MULTIPLEXER, AMX, OUTPUT LINES WITH THE RESPECTIVE FOUR BITS FROM THE CACHE ADDRESS MEMORY. TWELVE BITS OF THE ADDRESS ARE BROKEN DOWN THUS: BITS 10 THROUGH 13 FOR THE FIRST COMPARATOR; BITS 14 THROUGH 17 FOR THE NEXT; AND BITS 18 THROUGH 21 FOR THE LAST. THE METHOD CHOSEN FOR THIS TEST IS TO TAKE EACH POSSIBLE 4-BIT INPUT CONDITION FOR A COMPARATOR FROM THE ADDRESS MEMORY AND PUT EVERY POSSIBLE 4-BIT COMBINATION ON THE AMX SIDE OF THE COMPARATOR. FOR 4-BITS THERE ARE 16 (DEC) CONDITIONS. THUS FOR EVERY 4-BIT ADDRESS MEMORY INPUT TO THE COMPARATOR THERE ARE 16 AMX INPUT COMBINATIONS ONE OF WHICH WILL CAUSE A MATCH AND MAKE THE REFERENCE A HIT. THE OTHER 15 SHOULD OF COURSE BE MISSES.

TEST 14 CACHE ADDRESS MEMORY COUNT PATTERN TEST

THIS IS A TEST OF THE ADDRESS MEMORY IN THE CACHE. EVERY BIT IN THE MEMORY IS TURNED ON AND OFF WITHIN THE LIMITATIONS OF MEMORY SIZE. THE MANNER IN WHICH THIS IS DONE IS TO ATTEMPT TO MAKE EVERY ADDRESS IN AVAILABLE MEMORY A HIT IN EACH GROUP.

TEST 15 CACHE ADDRESS MEMORY PARITY LOGIC TEST

THIS IS A TEST OF THE PARITY CHECKERS AND PARITY GENERATOR OF THE CACHE ADDRESS MEMORY. EVERY POSSIBLE ADDRESS TAG, BITS 21 THROUGH 10, WHICH CAN BE STORED IN THE CACHE ADDRESS MEMORY IS GENERATED, MADE A HIT AND THE MAINTENANCE REGISTER IS THEN USED TO FORCE A CACHE ADDRESS MEMORY PARITY ERROR AT EACH OF THE ADDRESSES GENERATED. NOTE THAT BITS 9 THROUGH 0 OF THE ADDRESSES

IS NOT OF CONCERN, SO THESE BITS

WILL BE THE SAME FOR EACH ADDRESS; THIS IS BECAUSE ONLY BITS 21 THROUGH 10 ARE STORED IN THE ADDRESS MEMORY THEREFORE ONLY THESE BITS ARE PARITY CHECKED IN THE CACHE ADDRESS MEMORY PARITY CHECKERS. ALSO NOTE THAT THE RANGE OF THE ADDRESSES MUST BE LIMITED TO BETWEEN THE BOUNDS IMPOSED BY THE HIGHEST AVAILABLE MEMORY WORD AND THE LAST WORD OF MEMORY USED BY THIS PROGRAM. THE MANNER IN WHICH THE ERROR WILL BE FORCED WILL BE TO PUT THE INSTRUCTIONS:

```
        1$:     MOV     R4,(R2)
        TSTADS: CLR     (R2)
                RTS     PC
```

AT THE PARTICULAR ADDRESS BEING TESTED, WHERE 'TSTADS' IS THE ADDRESS BEING TESTED. R4 CONTAINS A PATTERN TO BE LOADED IN THE MAINTENANCE REGISTER WHICH WILL FORCE AN ERROR IN THE CACHE ADDRESS MEMORY; R2 CONTAINS THE ADDRESS OF THE MAINTENANCE REGISTER. NOTE FOR EACH ADDRESS R4 WILL FIRST BE SUCH AS TO CAUSE AN ERROR IN THE LOW BYTE ADDRESS PARITY CHECKER THEN AT THE SAME ADDRESS AN ERROR WILL BE FORCED ON THE HIGH BYTE! THE SEQUENCE OF TEST ADDRESSES WILL BE GENERATED TWICE ONCE MAKING THEM HITS IN GROUP 0 THEN MAKING THEM HITS IN GROUP 1.

TEST 16 CACHE ADDRESS MEMORY DUAL ADDRESS TEST, UPWARD

THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS MEMORY. AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS' ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP BEING TESTED. OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY ON THE SYSTEM. NOTE THAT HERE THE WORD 'TAG' REFERS TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21, WHICH ARE STORED IN THE CACHE ADDRESS MEMORY. HERE THE ADDRESS MEMORY IS WRITTEN IN THE UPWARD DIRECTION, THAT IS 'TAG' 1 IS WRITTEN FIRST, 'TAG' 2 SECOND ETC. THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED

TO SEE IF IT IS A HIT, THUS MAKING SURE NO 'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER 'TAG'. NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD INVOLVE WRITTING THE 'TAGS' IN THE DOWNWARD DIRECTION AS WELL AS THE UPWARD DIRECTION. THE DOWNWARD WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND IN TST13.

TEST 17 CACHE ADDRESS MEMORY DUAL ADDRESS TEST, DOWNWARD

THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS MEMORY. AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS' ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP BEING TESTED. OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY ON THE SYSTEM. NOTE THAT HERE THE WORD 'TAG' REFERS TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21, WHICH ARE STORED IN THE CACHE ADDRESS MEMORY. HERE THE ADDRESS MEMORY IS WRITTEN IN THE DOWNWARD DIRECTION, THAT IS 'TAG' 256 IS WRITTEN FIRST, 'TAG' 255 SECOND ETC. THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED TO SEE IF IT IS A HIT, THUS MAKING SURE NO 'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER 'TAG'. NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD INVOLVE WRITTING THE 'TAGS' IN THE UPWARD DIRECTION AS WELL AS THE DOWNWARD DIRECTION. THE UPWARD WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND IN TST12.

TEST 20 CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ONES TEST

THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC. THIS IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING A WRITE. IT DESIGNATES

WHICH BYTES OF THE TWO WORDS OF DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO BE WRITTEN. THIS WILL BE A TEST DOING CPU DATOB REFERENCES TO THE CACHE. THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN OF ZEROES.

TEST 21 CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ZEROES TEST

THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC. HERE CPU DATOB'S WILL MOVE ZEROES INTO A BACKROUND PATTERN OF ONES.

TEST 22 CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ONES TEST

THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC. THIS IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING A WRITE. IT DESIGNATES WHICH BYTES OF THE TWO WORDS OF DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO BE WRITTEN. THIS WILL BE A TEST DOING UNIBUS DATOB REFERENCES TO THE CACHE. THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN OF ZEROES.

TEST 23 CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ZEROES TEST

THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC. HERE UNIBUS DATOB'S WILL MOVE ZEROES INTO A BACKROUND PATTERN OF ONES.

TEST 24 CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST

THIS TEST IS EXECUTED OPTIONALLY, ON THE CONDITION THAT BIT 12 OF THE SWITCH REGISTER IS ON WHEN PROGRAM CONTROL REACHES THIS POINT. IF THIS SWITCH IS OFF THEN CONTROL IS PASSED TO THE NEXT TEST. THIS IS DONE BECAUSE THIS TEST REQUIRES OPERATOR INTERVENTION. THE USER IS ASKED TO

GO THROUGH A POWER DOWN-POWER UP SEQUENCE. THEN A SIMPLE SCAN IS MADE OF MEMORY WHICH CAUSES ALL DATA AND ADDRESS MEMORY LOCATIONS IN THE CACHE TO BE PARITY CHECKED. IF THE POWER UP-CACHE INVLIDATER LOGIC WORKED NO PARITY ERRORS CAN OCCUR. BUT IF THIS INVALIDATER FAILED THERE IS AN EXTREMELY HIGH PROBABILITY FOR THE OCCURENCE OF A CACHE DATA OR CACHE ADDRESS PARITY ERROR. IN FACT IF THE INVALIDATER CIRCUIT IS COMPLETELY INOPERATIVE IT WILL BE VIRTUALLY IMPOSSIBLE TO RESTART THE PROGRAM. WHEREAS MINOR OR NO FAILURES CAN AND WILL BE REPORTED. IF NO PARITY ERRORS ARE ENCOUNTERED THE USER WILL BE NOTIFIED SO THAT HE CAN KNOW IF A FATAL FAILURE HAS OCCURRED.

TEST 25 CACHE DATA MULTIPLEXER, CDMX, TEST

THIS TEST PUTS DIFFERENT PATTERNS OF DATA AT THE INPUTS OF THE CDMX AND TESTS FOR PROPER SELECTION AND GOOD DATA.

TEST 26 CACHE DATA MEMORY ADDRESS DRIVERS TEST

THIS TEST PERFORMS A DUAL ADDRESS TEST ON THE CACHE DATA MEMORIES OF BOTH GROUPS.

TEST 27 CACHE DATA MEMORY COUNT PATTERN TEST

THIS TEST RUNS A COUNT PATTERN THROUGH EACH LOCATION OF THE CACHE DATA MEMORY FOR EACH GROUP.

TEST 30 CACHE DATA MEMORY PARITY CHECKERS LOW BYTE TEST

THIS IS A TEST OF THE TWO CACHE DATA MEMORY PARITY CHECKERS FOR THE LOW BYTE, ONE FOR EACH GROUP. THE MAINTENANCE REGISTER ISUSED TO FORCE A PARITY A PARITY ERROR AT EVERY DATA PATTERN WHICH HAS A ONE PARITY

BIT. NOTE THAT THE CACHE DATA MEMORY PARITY HAS, EFFECTIVELY, ODD PARITY. THE MAINTENANCE FUNCTION ON THE CACHE DATA MEMORY PARITY CHECKERS HAS THE EFFECT OF FORCING THE PARITY BIT OF THE BYTE BEING CHECKED TO ZERO. THIS MEANS THAT ONCE THIS MAINTENANCE FUNCTION IS ENABLED THE ERROR WILL OCCUR ON A SUBSEQUENT READ OF A BYTE WITH A ONE PARITY BIT, THAT IS BYTES WITH ZERO PARITY BITS WILL NOT CAUSE THE ERROR.

TEST 31 CACHE DATA MEMORY PARITY CHECKERS HIGH BYTE TEST

THIS IS A TEST OF THE TWO CACHE DATA MEMORY PARITY CHECKERS FOR THE HIGH BYTE, ONE FOR EACH GROUP. THE MAINTENANCE REGISTER ISUSED TO FORCE A PARITY A PARITY ERROR AT EVERY DATA PATTERN WHICH HAS A ONE PARITY BIT. NOTE THAT THE CACHE DATA MEMORY PARITY HAS, EFFECTIVELY, ODD PARITY. THE MAINTENANCE FUNCTION ON THE CACHE DATA MEMORY PARITY CHECKERS HAS THE EFFECT OF FORCING THE PARITY BIT OF THE BYTE BEING CHECKED TO ZERO. THIS MEANS THAT ONCE THIS MAINTENANCE FUNCTION IS ENABLED THE ERROR WILL OCCUR ON A SUBSEQUENT READ OF A BYTE WITH A ONE PARITY BIT, THAT IS BYTES WITH ZERO PARITY BITS WILL NOT CAUSE THE ERROR.

TEST 32 CACHE DATA MEMORY WORST CASE NOISE TEST

THIS TEST DOES A GALLOPING 0'S AND 1'S OR PING PONG TEST ON THE CACHE BIPOLAR DATA MEMORY.

TEST 33 CACHE DATA MEMORY CHIP SELECTION LOGIC TEST

THIS ROUTINE TESTS THE 'CHIP-SET' ENABLE LOGIC FOR THE CACHE DATA MEMORY. TO DEFINE THE TERM 'CHIP-SET' CONSIDER THE CACHE MEMORY

AS BEING DIVIDED INTO FOUR SETS OF 256 (DEC) X 1 BIT BIPOLAR MEMORY CHIPS. EACH SET IS MADE UP OF 18 CHIPS, THE 745200, EACH CHIP REPRESENTS ONE BIT OF DATA OR PARITY, THUS 16 DATA BITS PLUS TWO PARITY BITS CORRESPOND TO THE 18 CHIPS IN EACH GROUP. THE 'CHIP-SETS' THEN CORRESPOND TO THE STRUCTURE OF THE MEMORY IN THIS WAY:

```
        SET 0    GROUP 0 EVEN WORD
        SET 1    GROUP 0 ODD WORD
        SET 2    GROUP 1 EVEN WORD
        SET 3    GROUP 1 ODD  WORD  A
```

DIFFERENT PATTERN, 000000 177777 125252 AND 052525, IS WRITTEN INTO EACH GROUP AND THEN READ BACK. EVERY PERMUTATION OF THE

FOUR TEST PATTERNS IN THE FOUR SETS IS TRIED AND CHECKED. FOR EACH PERMUTATION OF THE TEST PATTERNS THIS ROUTINE FIRST WRITES 'UP' (SET 0 FIRST THEN 1,2 AND 3) THEN 'DOWN' (SET 3 FIRST THEN 2,1 AND 0).

TEST 34 CACHE DATA MEMORY BYTE ENABLE LOGIC TEST

THIS TEST PERFORMS A CHECK OF THE BYTE ENABLE LOGIC IN THE CACHE DATA MEMORY. THE BYTE PATTERNS 1, 2, 4, 10, 20, 40, 100 A 200 ARE USED. THE FIRST FOUR PATTERNS ARE WRITTEN IN CONSECUTIVE BYTE LOCATIONS WHICH ARE HITS IN GROUP 0. THE REMAINING FOUR PATTERNS ARE WRITTEN IN CONSECUTIVE BYTE LOCATIONS WHICH ARE HITS IN GROUP 1. EACH PATTERN IS READ BACK CHECKED AND THE COMPLIMENT PATTERN IS WRITTEN. AFTER ALL THE PATTERNS HAVE BEEN CHECKED AND COMPLEMENTED THE COMPLIMENTED PATTERNS ARE CHECKED.

TEST 35 CACHE ARBITRATION AND HIGH SPEED I/O TEST

```
THIS IS A TEST OF:
        1.      CACHE ARBITRATION
        2.      THE  MASS  BUS   AND
        UNIBUS PORTS TO THE CACHE
        3.      HIGH   SPEED    I/O
        THROUGH THE CACHE


IT MAKE USE   OF    THE    FOLLOWING
        DEVICES:
        1.      RS04
        2.      RP04
        3.      RK05
        4.      MASS BUSS TESTER
        5.      UNIBUS EXERCISER

IF ANY OF THESE DEVICES ARE  PRESENT
AND WRITE  ENABLED THEY WILL BE USED
IN THIS TEST.  ONLY THE LOWEST WRITE
ENABLED  DRIVE NUMBER OF EACH DEVICE
WILL BE USED.

        CAUTION!!!  THIS  TEST  WILL
        WRITE  ON THE DISKS IT USES.
        SO   VITAL   SYSTEMS   DISKS
        SHOULD  BE  REMOVED OR WRITE
        PROTECTED   BEFORE   RUNNING
        THIS DIAGNOSTIC.

IF UNIT ZERO OF A PARTICULAR  DEVICE
IS  WRITE  PROTECTED  THEN THIS TEST
WILL TRY TO USE UNIT ONE, ETC.

ALL AVAILABLE  DEVICES  ARE  STARTED
DOING  TRANSFERS AT THE SAME TIME TO
DIFFERENT  PARTS  OF  MEMORY.   EACH
DEVICE  HAS  A CONTROL ROUTINE WHICH
DRIVES  THAT  DEVICE   THROUGH   THE
CYCLE:
        1.      WRITE A RANDOM  DATA
        PATTERN IN MEMORY
        2.      COPY  THAT   PATTERN
        ONTO THE DISK
        3.      WRITE CHECK THE DISK

        4.      READ THE PATTERN OFF
        THE DISK BACK INTO MEMORY
        5.      CHECK DATA
        6.      START OVER AT 1.

EACH DEVICE IS CAUSED TO GO  THRUUGH
THIS CYCLE A PREDETERMINED NUMBER OF
TIMES.  THIS NUMBER IS CONTAINED  IN
THE  LOCATION,  CYCNT,  AND  CAN  BE
CHANGED BY THE USER AT  THE  CONSOLE
```

TO ANY VALUE HE DESIRES.

INTERRUPTS ARE ENABLED SO THAT IT IS POSSIBLE TO GET MANY DEVICES DOING TRANSFERS AT ONCE.

UNFORTUNATELY THE DEGREE TO WHICH FAULTS CAN BE ISOLATED IS LIMITED BY THE FACT THAT THERE ARE MANY ELEMENTS, DEVICES, INVOLVED. THESE ERRORS ARE REPORTED:

1. ALL DEVICE ERRORS
2. ALL DATA OR PARITY ERRORS

NOTE THAT THIS NOT INTENDED TO BE USED AS AN I/O DEVICE DIAGNOSTIC! ALL THE DEVICES WHICH ARE USED ARE ASSUMED TO BE IN PROPER WORKING CONDITION.

TEST 36 MASS BUS CACHE WRITE HIT CYCLE, INVALIDATION TEST

THIS IS A TEST OF CACHE INVALIDATION ON MASS BUS CYCLES WHICH ARE WRITE HITS IN THE CACHE. A GROUP OF LOCATIONS IS MADE HITS AND THEN A MASS BUS DEVICE IS CALLED UPON TO DO TRANSFERS, WRITES TO THOSE LOCATIONS. THOSE WRITES SHOULD THUS BE INVALIDATED.

NOTE: THE FOLLOWING TESTS ARE EXECUTED ON A KB11-CM ONLY!

TEST 37 CHECK IVSS, VSIU BITS

THIS TEST CHECKS THAT THE IVSS AND VSIU BITS OF THE CACHE CONTROL REGISTER CAN BE SET AND CLEARED. VCIP IS ALSO CHECKED.

TEST 40 CHECK VSIU BIT, WITH IVSS ALREADY SET

THIS TEST CHECKS THAT THE ''VALID STORE IN USE'' (VISU) BIT CAN BE SET AND CLEARED WHEN THE IVSS IS ALREADY SET.

TEST 41 CHECK VCIP SETS WHEN CF IS SET

THIS TEST CHECKS THAT THE VCIP SETS WHEN CACHE-FLUSH IS DONE AND IT CLEARS OUT WITHIN A CERTAIN TIME AFTER THE FLUSH OF VALID STORE IS OVER

TEST 42 CHECK CACHE FLUSH & VALID STORE SWITCHING

THIS TEST CHECKS THAT WHEN A CACHE FLUSH IS DONE BY SETTING CF IN CCR, THE VALID STORE IN USE (VSIU) SURTCHES. VALID STORE SWITCHING FROM STORE-A TO STORE-B AND VICE-VERSA IS CHECKED

TEST 43 CHECK IVSS INHIBITS SWITCHING OF VALID STORE IN USE

THIS TEST CHECKS THAT WHEN "INHIBIT VALID STORE SWITCHING" (IVSS) IS SET AND FLUSH-CACHE BIT IS SET, THE VALID STORE IN USE DOES NOT SWITCH

TEST 44 CHECK VALID STORES (A & B) FOR GROUP 0

THIS TEST CHECKS THE TWO VALID STORES (A&B) FOR GROUP 0 OF THE CACHE. WHEN A CACHE-FLUSH IS ISSUED, THE CACHE SHOULD BE INVALIDATED BY SWITCHING THE VALID STORE IN USE THE TEST-CODE IS MADE HIT IN GROUP 1 (WHICH IS NOT BEING TESTED). THE TEST DATA IS MADE HIT IN GROUP 0. FLUSH-CACHE BIT IS SET IN THE CCR. IT IS CHECKED THAT THE TEST-DATA WHICH WAS HIT (MADE PREVIOUSLY) IN GROUP 0 IS NO MORE A HIT. EACH LOCATION OF THE TEST-DATA BLOCK IS REFERENCED AND CHECKED IF IT WAS A MISS. OTHERWISE AN ERROR IS REPORTED. AS A RESULT OF THE CACHE FLUSH THE VALID STORE SHOULD HAVE SWITCHED FROM 0 TO 1. THEN THE VALID STORE IS FORCED TO BE 0 AND THE TEST-DATA IS REFERENCED AGAIN. IT IS CHECKED IF IT WAS A MISS.

TEST 45 CHECK VALID STORES (A&B) FOR GROUPES 0 & 1

THIS TEST CHECKS THAT HIT CAN BE OBTAINED FROM BOTH GROUPS (0&1) OF THE CACHE, FROM EACH OF THE TWO VALID STORES (A&B) PER GROUP. THUS ALL 4 VALID STORES GET CHECKED. TEST-DATA (UNIQUE) IS MADE A HIT IN GROUP 0 USING THE FIRST VALID STORE A. TEST-CODE IS MADE A HIT IN THE GROUP NOT BEING TESTED. TEST-DATA IS READ BACK AND CHECKED FOR CORRECTNESS. IT IS ALSO CHECKED IF THE TEST-DATA REFERENCE WAS A HIT. THE TESTING IS REPEATED FOR VALID STORE B. THE ENTIRE TEST (ABOVE) IS REPEATED FOR GROUP 1.

TEST 46 CHECK VALID STORES (A &B) FOR GROUP 1

THIS TEST CHECKS RTHE TWO VALID STORES (A&B) FOR GROUP 1 OF THE CACHE. WHEN A CACHE-FLUSH

IS ISSUED, THE CACHE SHOULD BE INVALIDATED BY SWITCHING THE VALID STORE IN USE. THE TEST-CODE IS MADE HIT IN GROUP 1 (WHICH IS NOT BEING TESTED). THE TEST DATA IS MADE HIT IN GROUP 0. FLUSH-CACHE HIT IS SET IN THE CCR. IT IS CHECKED THAT THE TEST-DATA WHICH WAS HIT (MADE PRECIOUSLY) IN GROUP 0 IS NO MORE A HIT, EACH LOCATION OF THE TEST-DATA BLOCK IS REFERENCED AND CHECKED IF IT WAS A MISS. OTHERWISE AN ERROR IS REPORTED. AS A RESULT OF THE CACHE FLUSH THE VALID STORE SHOULD HAVE SWITCHED FROM 0 TO 1. THEN THE VALID STORE IS FORCED TO BE 0 AND THE TEST-DATA IS REFERENCED AGAIN. IT IS CHECKED IF IT WAS A MISS. THE WHOLE TEST IS REPEATED USING VALID-STORE B (1).

TEST 47 CHECK CACHE TURNS OFF WHEN FLUSH IS DONE WITH IVSS SET

THIS TEST CHECKS THAT IF CACHE-FLUSH IS DONE (SETTING CF), WHEN IVSS IS SET, THE VALID STORES ARE NOT SWITCHED AND THE CACHE IS TURNED OFF (AND A SLOW FLUSH IS PERFORMED). THUS, ANY REFERENCE TO A PREVIOUSLY CACHED DATA SHOULD RESULT IN CACHE MISS. TEST-DATA IS MADE HIT IN GROUP 0 (BEING TESTED). TEST CODE IS MADE HIT IN GROUP 1.IVSS IS SET AND A FLUSH IS DONE. PREVIOUSLY CACHED TEST-DATA IS REFERENCED TO CHECK IT IS A MISS. THE TEST IS REPEATED FOR BOTH GROUPS AND VALID STORES.

TEST 50 CHECK CACHE TURNS OFF ON A BACK-TO-BACK FLUSH

THIS TEST CHECKS THAT THE CACHE TURNS OFF AND FORCES ALL REFERENCES TO THE MAIN MEMORY WHEN BACK-TO-BACK CACHE FLUSHES ARE DONE. WHEN A CACHE FLUSH IS INITIATED WHILE THE PREVIOUS ONE IS IN PROGRESS, IT IS KNOWN AS BACK-TO-BACK FLUSH.

TEST 51 CHECK CACHE-BYPASS

THIS TEST CHECKS THE CACHE BYPASS FUNCTION. WHEN THE 'BYPASS CACHE'' IS SET IN THE CACHE CONTROL REGISTER ALL REFERENCES ARE FORCED TO MAIN MEMORY. IF A READ OR WRITE HIT OCCURS THAT LOCATION IS INVAL- -IDATED IN THE TAG STORE. FIRST, THE TEST CODE IS MADE HIT IN GROUP 1 BY FORCE-REPLACING GROUP 1. THEN THE TEST-DATA IS MADE HIT IN GROUP 0. CACHE-BYPASS IS SET AND THE TEST DATA (WHICH HAS BEEN CACHED IN GROUP 0) IS REFERENCED. THE REFERENCES ARE CHECKED FOR MISSES (THE

TEST-DATA INSIDE THE CACHE GROUP-0 SHOULD HAVE BEEN INVALIDATED WHEN REFERENCES WERE MADE WITH CACHE-BYPASS SET.) THE ENTIRE TEST IS REPEATED, SELECTING THE OTHER VALID STORE AND THEN WITH TEST-DATA IN GROUP 1.

TEST 52 CHECK CACHE IS BYPASSED ON ASRB OPERAND

THIS TEST CHECKS THAT THE CACHE IS BYPASSED ON THE OPERAND OF THE ASRB INSTRUCTION AND ALSO THE OPERAND IS INVALIDATED. TEST-CODE (INCLDING THE OPERAND OF THE ASRB) IS MADE HIT IN GROUP 1. THEN ASRB INSTRUCTION IS EXECUTED ON THE CACHED OPERAND. IT IS CHECKED IF THE REFERENCE TO THE BYTE-OPERAND WAS A MISS. THEN THE SAME OPERAND REFERENCED USING AN ORDINARY (NON-BYPASSING) INSTRUCTED. AGAIN, THE REFERENCE IS CHECKED FOR A MISS.

TEST 53 CHECK CACHE VALID STORE PARITY CHECKER

THIS TEST FORCES VALID STORE PARITY ERROR IN THE FOUR VALID STORES AND CHECKS THE PARITY CHECKERS.

TEST 54 CHECK THAT CACHE-MISS OCCURS ON A VALID STORE PARITY ERROR

THIS TEST FORCES A VALID STORE PARITY ERROR AND CHECKS THAT A MISS OCCURS ON THE REFERENCE THAT CAUSED THE PARITY ERROR. THE CACHE LOCATION THAT GAVE THE PARITY ERROR IS INVALIDATED AND A SLOW CYCLE IS PERFORMED TO THE MAIN MEMORY. THIS TEST IS PERFORMED WITH THE ''DISABLE TRAPS'' BIT OF THE CACHE CONTROL REGISTER SET, THUS A PARITY ERROR TRAP WILL NOT OCCUR. THIS IS DONE SO THAT THE HIT-MISS REGISTER CAN BE READ WITHOUT LOSING THE INFORMATION CONTAINED IN IT.

TEST 55 CHECK BYP ON KERNEL PAGE BITS
THIS TEST IS EXECUTED ONLY ON KB11-E/EM/CM

TEST 56 CHECK BYP ON SUPERVISOR PAGE BITS
THIS TEST IS EXECUTED ONLY ON KB11-E/EM/CM

TEST 57 CHECK BYP ON USER PAGE BITS
THIS TEST IS EXECUTED ONLY ON KB11-E/EM/CM

TEST 60 CHECK CACHE BYPASS ON VIRTUAL PAGE
THIS TEST IS EXECUTED ONLY ON KB11-E/EM/CM

```
                        .TITLE  CEKBD-D PDP 11/70-74MP CACHE DIAGNOSTIC PART 2
                        ;*COPYRIGHT (C) 1975, 1978
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ANTHONY S. VEZZA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-A5-1).
                        ;*
        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
                        ;*        12                  EXECUTE THE POWER UP INVALIDATOR TEST
                        ;*        11                  INHIBIT ITERATIONS
                        ;*        10                  BELL ON ERROR
                        ;*         9                  LOOP ON ERROR
                        ;*         8                  LOOP ON TEST IN SWR<6:0>
                        ;*         7                  SKIP EXECUTION OF TESTS WHICH 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
000004                  R4=     %4                       ;;GENERAL REGISTER
000005                  R5=     %5                       ;;GENERAL REGISTER
000006                  R6=     %6                       ;;GENERAL REGISTER
000007                  R7=     %7                       ;;GENERAL REGISTER
                        .EQUIV  R0,R10                   ;;GENERAL REGISTER
                        .EQUIV  R1,R11                   ;;GENERAL REGISTER
                        .EQUIV  R2,R12                   ;;GENERAL REGISTER
                        .EQUIV  R3,R13                   ;;GENERAL REGISTER
                        .EQUIV  R4,R14                   ;;GENERAL REGISTER
                        .EQUIV  R5,R15                   ;;GENERAL REGISTER
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
        040000                       BIT14=  40000
        020000                       BIT13=  20000
        010000                       BIT12=  10000
        004000                       BIT11=  4000
        002000                       BIT10=  2000
        001000                       BIT09=  1000
        000400                       BIT08=  400
        000200                       BIT07=  200
        000100                       BIT06=  100
        000040                       BIT05=  40
        000020                       BIT04=  20
        000010                       BIT03=  10
        000004                       BIT02=  4
        000002                       BIT01=  2
        000001                       BIT00=  1
                                     .EQUIV  BIT09,BIT9
                                     .EQUIV  BIT08,BIT8
                                     .EQUIV  BIT07,BIT7
                                     .EQUIV  BIT06,BIT6
                                     .EQUIV  BIT05,BIT5
                                     .EQUIV  BIT04,BIT4
                                     .EQUIV  BIT03,BIT3
                                     .EQUIV  BIT02,BIT2
                                     .EQUIV  BIT01,BIT1
                                     .EQUIV  BIT00,BIT0

                                     ;*BASIC ''CPU'' TRAP VECTOR ADDRESSES
        000004                       ERRVEC= 4                 ;;TIME OUT AND OTHER ERRORS
        000010                       RESVEC= 10                ;;RESERVED AND ILLEGAL INSTRUCTIONS
        000014                       TBITVEC=14                ;;''T'' BIT
        000014                       TRTVEC= 14                ;;TRACE TRAP
        000014                       BPTVEC= 14                ;;BREAKPOINT TRAP (BPT)
        000020                       IOTVEC= 20                ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
        000024                       PWRVEC= 24                ;;POWER FAIL
        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
```

```


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




                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS


                        ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

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

                        ;*USER "I" PAGE DESCRIPTOR REGISTERS

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

                        ;*USER "D" PAGE DESCRIPTOR REGISTORS

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

                        ;*USER "I" PAGE ADDRESS REGISTERS

177640                  UIPAR0= 177640
177642                  UIPAR1= 177642
177644                  UIPAR2= 177644
177646                  UIPAR3= 177646
177650                  UIPAR4= 177650
```

```
        177652                  UIPAR5= 177652
        177654                  UIPAR6= 177654
        177656                  UIPAR7= 177656

                                ;*USER 'D'' PAGE ADDRESS REGISTERS

        177660                  UDPAR0= 177660
        177662                  UDPAR1= 177662
        177664                  UDPAR2= 177664
        177666                  UDPAR3= 177666
        177670                  UDPAR4= 177670
        177672                  UDPAR5= 177672
        177674                  UDPAR6= 177674
        177676                  UDPAR7= 177676

                                ;*SUPERVISOR ''I'' PAGE DESCRIPTOR REGISTERS

        172200                  SIPDR0= 172200
        172202                  SIPDR1= 172202
        172204                  SIPDR2= 172204
        172206                  SIPDR3= 172206
        172210                  SIPDR4= 172210
        172212                  SIPDR5= 172212
        172214                  SIPDR6= 172214
        172216                  SIPDR7= 172216

                                ;*SUPERVISOR 'D'' PAGE DESCRIPTOR REGISTERS

        172220                  SDPDR0= 172220
        172222                  SDPDR1= 172222
        172224                  SDPDR2= 172224
        172226                  SDPDR3= 172226
        172230                  SDPDR4= 172230
        172232                  SDPDR5= 172232
        172234                  SDPDR6= 172234
        172236                  SDPDR7= 172236

                                ;*SUPERVISOR ''I'' PAGE ADDRESS REGISTERS

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

                                ;*SUPERVISOR 'D'' PAGE ADDRESS REGISTERS

        172260                  SDPAR0= 172260
        172262                  SDPAR1= 172262
        172264                  SDPAR2= 172264
        172266                  SDPAR3= 172266
        172270                  SDPAR4= 172270
        172272                  SDPAR5= 172272
```

```
          172274                      SDPAR6= 172274
          172276                      SDPAR7= 172276

                                      ;*KERNEL ''I'' PAGE DESCRIPTOR REGISTERS

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

                                      ;*KERNEL 'D'' PAGE DESCRIPTOR REGISTERS

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

                                      ;*KERNEL ''I'' PAGE ADDRESS REGISTERS

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

                                      ;*KERNEL 'D'' PAGE ADDRESS REGISTERS

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




                                      .SBTTL  UNIBUS MAP REGISTER DEFINITIONS


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

```

```

      170200                  MAPL00 = 170200
      170202                  MAPH00 = 170202
      170204                  MAPL01 = 170204
      170206                  MAPH01 = 170206
      170210                  MAPL02 = 170210
      170212                  MAPH02 = 170212
      170214                  MAPL03 = 170214
      170216                  MAPH03 = 170216
      170220                  MAPL04 = 170220
      170222                  MAPH04 = 170222
      170224                  MAPL05 = 170224
      170226                  MAPH05 = 170226
      170230                  MAPL06 = 170230
      170232                  MAPH06 = 170232
      170234                  MAPL07 = 170234
      170236                  MAPH07 = 170236
      170240                  MAPL10 = 170240
      170242                  MAPH10 = 170242
      170244                  MAPL11 = 170244
      170246                  MAPH11 = 170246
      170250                  MAPL12 = 170250
      170252                  MAPH12 = 170252
      170254                  MAPL13 = 170254
      170256                  MAPH13 = 170256
      170260                  MAPL14 = 170260
      170262                  MAPH14 = 170262
      170264                  MAPL15 = 170264
      170266                  MAPH15 = 170266
      170270                  MAPL16 = 170270
      170272                  MAPH16 = 170272
      170274                  MAPL17 = 170274
      170276                  MAPH17 = 170276
      170300                  MAPL20 = 170300
      170302                  MAPH20 = 170302
      170304                  MAPL21 = 170304
      170306                  MAPH21 = 170306
      170310                  MAPL22 = 170310
      170312                  MAPH22 = 170312
      170314                  MAPL23 = 170314
      170316                  MAPH23 = 170316
      170320                  MAPL24 = 170320
      170320                  MAPH24 = 170320
      170324                  MAPL25 = 170324
      170326                  MAPH25 = 170326
      170330                  MAPL26 = 170330
      170332                  MAPH26 = 170332
      170334                  MAPL27 = 170334
      170336                  MAPH27 = 170336
      170340                  MAPL30 = 170340
      170342                  MAPH30 = 170342
      170344                  MAPL31 = 170344
      170346                  MAPH31 = 170346
      170350                  MAPL32 = 170350
      170352                  MAPH32 = 170352
      170354                  MAPL33 = 170354
```

```
170356                  MAPH33 = 170356
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





                                ;DEFINITIONS

100000                          VSPE=BIT15
040000                          IVSS=BIT14
020000                          VSIU=BIT13
010000                          VCIP=BIT12
004000                          DMMA=BIT11
002000                          FVPE=BIT10
001000                          UCB=BIT9
000400                          FCAC=BIT8
000040                          S1=BIT5
000020                          S0=BIT4
000010                          M1=BIT3
000004                          M0=BIT2
000002                          DUT=BIT1
000001                          DT=BIT0

100000                          BYP=BIT15

000054                          S1M0M1=BIT5+BIT3+BIT2
000034                          S0M0M1=BIT4+BIT3+BIT2
000014                          M0M1=BIT3+BIT2

177746                          CONTRL=177746
177752                          HITMIS=177752
177744                          MSER=177744
```

```








        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
        001500                  STACK=1500

                                .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  004112                  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  051320                          .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.

        001500                                  .=1500

001500                                  $CMTAG:                                 ;;START OF COMMON TAGS
001500  000000                          $PASS:  .WORD   0                       ;;CONTAINS PASS COUNT
001502     000     000     000          $TSTNM: .BYTE   0,0,0                   ;;CONTAINS THE TEST NUMBER
001505     000                          $ERFLG: .BYTE   0                       ;;CONTAINS ERROR FLAG
001506  000000                          $ICNT:  .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
001510  000000                          $LPADR: .WORD   0                       ;;CONTAINS SCOPE LOOP 1500
001512  000000                          $LPERR: .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
001514  000000                          $ERTTL: .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
001516     000                          $ITEMB: .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
001517     001                          $ERMAX: .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
001520  000000                          $ERRPC: .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001522  000000                          $GDADR: .WORD   0                       ;;CONTAINS 1500 OF 'GOOD' DATA
001524  000000                          $BDADR: .WORD   0                       ;;CONTAINS 1500 OF 'BAD' DATA
001526  000000                          $GDDAT: .WORD   0                       ;;CONTAINS 'GOOD' DATA
001530  000000                          $BDDAT: .WORD   0                       ;;CONTAINS 'BAD' DATA
001532  000000  000000  000000                  .WORD   0,0,0                   ;;RESERVED--NOT TO BE USED
001540  177560                          $TKS:   177560                          ;;TTY KBD STATUS
001542  177562                          $TKB:   177562                          ;;TTY KBD BUFFER
001544  177564                          $TPS:   177564                          ;;TTY PRINTER STATUS REG. 1500
001546  177566                          $TPB:   177566                          ;;TTY PRINTER BUFFER REG. 1500
001550     000                          $NULL:  .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
001551     002                          $FILLS: .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001552     012                          $FILLC: .BYTE   12                      ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
001553     000                          $TPFLG: .BYTE   0                       ;;''TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
001554  000000                          $REGAD: .WORD   0                       ;;CONTAINS THE 1500 FROM
                                                                                ;;WHICH  ($REG0) WAS OBTAINED
001556  000000                          $REG0:  .WORD   0                       ;;CONTAINS (($REGAD)+0)
001560  000000                          $REG1:  .WORD   0                       ;;CONTAINS (($REGAD)+2)
001562  000000                          $REG2:  .WORD   0                       ;;CONTAINS (($REGAD)+4)
001564  000000                          $REG3:  .WORD   0                       ;;CONTAINS (($REGAD)+6)
001566  000000                          $REG4:  .WORD   0                       ;;CONTAINS (($REGAD)+10)
001570  000000                          $REG5:  .WORD   0                       ;;CONTAINS (($REGAD)+12)
001572  000000                          $REG6:  .WORD   0                       ;;CONTAINS (($REGAD)+14)
001574  000000                          $REG7:  .WORD   0                       ;;CONTAINS (($REGAD)+16)
001576  000000                          $REG10: .WORD   0                       ;;CONTAINS (($REGAD)+20)
001600  000000                          $REG11: .WORD   0                       ;;CONTAINS (($REGAD)+22)
001602  000000                          $REG12: .WORD   0                       ;;CONTAINS (($REGAD)+24)
001604  000000                          $REG13: .WORD   0                       ;;CONTAINS (($REGAD)+26)
001606  000000                          $REG14: .WORD   0                       ;;CONTAINS (($REGAD)+30)
001610  000000                          $REG15: .WORD   0                       ;;CONTAINS (($REGAD)+32)
001612  000000                          $REG16: .WORD   0                       ;;CONTAINS (($REGAD)+34)
001614  000000                          $REG17: .WORD   0                       ;;CONTAINS (($REGAD)+36)
001616  000000                          $REG20: .WORD   0                       ;;CONTAINS (($REGAD)+40)
001620  000000                          $REG21: .WORD   0                       ;;CONTAINS (($REGAD)+42)
001622  000000                          $REG22: .WORD   0                       ;;CONTAINS (($REGAD)+44)
001624  000000                          $REG23: .WORD   0                       ;;CONTAINS (($REGAD)+46)
001626  000000                          $TMP0:  .WORD   0                       ;;USER DEFINED
```

```
001630  000000                  $TMP1:   .WORD   0                  ;;USER DEFINED
001632  000000                  $TMP2:   .WORD   0                  ;;USER DEFINED
001634  000000                  $TMP3:   .WORD   0                  ;;USER DEFINED
001636  000000                  $TMP4:   .WORD   0                  ;;USER DEFINED
001640  000000                  $TMP5:   .WORD   0                  ;;USER DEFINED
001642  000000                  $TMP6:   .WORD   0                  ;;USER DEFINED
001644  000C00                  $TMP7:   .WORD   0                  ;;USER DEFINED
001646  000000                  $TMP10:  .WORD   0                  ;;USER DEFINED
001650  000000                  $TMP11:  .WORD   0                  ;;USER DEFINED
001652  000000                  $TMP12:  .WORD   0                  ;;USER DEFINED
001654  000000                  $TMP13:  .WORD   0                  ;;USER DEFINED
001656  000000                  $TMP14:  .WORD   0                  ;;USER DEFINED
001660  000000                  $TMP15:  .WORD   0                  ;;USER DEFINED
001662  000000                  $TMP16:  .WORD   0                  ;;USER DEFINED
001664  000000                  $TMP17:  .WORD   0                  ;;USER DEFINED
001666  000000                  $TMP20:  .WORD   0                  ;;USER DEFINED
001670  000000                  $TMP21:  .WORD   0                  ;;USER DEFINED
001672  000000                  $TMP22:  .WORD   0                  ;;USER DEFINED
001674  000000                  $TMP23:  .WORD   0                  ;;USER DEFINED
001676  000000                  $TIMES: 0                           ;;MAX. NUMBER OF ITERATIONS
001700  000000                  $ESCAPE:0                           ;;ESCAPE ON ERROR 1500
001702  177607  000377          $BELL:   .ASCIZ  <207><377><377>    ;;CODE FOR BELL
001706     077                  $QUES:   .ASCII  /?/                ;;QUESTION MARK
001707     015                  $CRLF:   .ASCII  <15>               ;;CARRIAGE RETURN
001710  000012                  $LF:     .ASCIZ  <12>               ;;LINE FEED
001712     000                  KB11E:   .BYTE   0          ;1174 WITHOUT MP CACHE FLAG
001713     000                  KB11EM:  .BYTE   0          ;1174 WITH MP CACHE FLAG
001714     000                  KB11CM:  .BYTE   0          ;KB11CM FLAG (1170 WITH MP MODS)
001715     000                  CISP:    .BYTE   0          ;CISP OPTION PRESENT FLAG

                                ;OPCODE FOR MFPT INSTRUCTION (AVAILABLE ON KB11-E AND KB11-EM ONLY)
        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


 001716                                 $ERRTB:


                                        ;ERROR TABLE FOR ERROR TYPE OUT:
                                        ;ITEM 1
 001716  070420  107657  113442                 .WORD   EM1,DH1,DT1,DF1
 001724  113117
                                        ;ITEM 2
 001726  070505  107732  113454                 .WORD   EM2,DH2,DT2,DF2
 001734  113123
                                        ;ITEM 3
 001736  070677  107732  113454                 .WORD   EM3,DH3,DT3,DF3
 001744  113123
                                        ;ITEM 4
 001746  071013  107732  113454                 .WORD   EM4,DH4,DT4,DF4
 001754  113123
                                        ;ITEM 5
 001756  071126  110033  113472                 .WORD   EM5,DH5,DT5,DF5
 001764  113131
                                        ;ITEM 6
 001766  071206  110033  113472                 .WORD   EM6,DH6,DT6,DF6
 001774  113131
                                        ;ITEM 7
 001776  071266  110033  113472                 .WORD   EM7,DH7,DT7,DF7
 002004  113131
                                        ;ITEM 10
 002006  071360  110033  113472                 .WORD   EM10,DH10,DT10,DF10
 002014  113131
                                        ;ITEM 11
 002016  071451  110065  113514                 .WORD   EM11,DH11,DT11,DF11
 002024  113141
                                        ;ITEM 12
 002026  071537  110141  113540                 .WORD   EM12,DH12,DT12,DF12
 002034  113152
                                        ;ITEM 13
 002036  071664  110234  113556                 .WORD   EM13,DH13,DT13,DF13
 002044  113160
                                        ;ITEM 14
 002046  071744  110307  113572                 .WORD   EM14,DH14,DT14,DF14
 002054  113165
```

```
                                       ;ITEM 15
 002056  072003  110402  113606                  .WORD     EM15,DH15,DT15,DF15
 002064  113172
                                       ;ITEM 16
 002066  072053  110426  113614                  .WORD     EM16,DH16,DT16,DF16
 002074  113174
                                       ;ITEM 17
 002076  072127  110514  113632                  .WORD     EM17,DH17,DT17,DF17
 002104  113202
                                       ;ITEM 20
 002106  072127  110514  113714                  .WORD     EM20,DH20,DT20,DF20
 002114  113202
                                       ;ITEM 21
 002116  072210  110574  113776                  .WORD     EM21,DH21,DT21,DF21
 002124  113232
                                       ;ITEM 22
 002126  072274  110647  114054                  .WORD     EM22,DH22,DT22,DF22
 002134  113260
                                       ;ITEM 23
 002136  072510  110716  114064                  .WORD     EM23,DH23,DT23,DF23
 002144  113263
                                       ;ITEM 24
 002146  072274  110647  114100                  .WORD     EM24,DH24,DT24,DF24
 002154  113260
                                       ;ITEM 25
 002156  072510  110716  114110                  .WORD     EM25,DH25,DT25,DF25
 002164  113263
                                       ;ITEM 26
 002166  072644  111006  114124                  .WORD     EM26,DH26,DT26,DF26
 002174  113270
                                       ;ITEM 27
 002176  073011  111053  114136                  .WORD     EM27,DH27,DT27,DF27
 002204  113274
                                       ;ITEM 30
 002206  072274  110647  114152                  .WORD     EM30,DH30,DT30,DF30
 002214  113260
                                       ;ITEM 31
 002216  073156  110716  114162                  .WORD     EM31,DH31,DT31,DF31
 002224  113263
                                       ;ITEM 32
 002226  072274  110647  114176                  .WORD     EM32,DH32,DT32,DF32
 002234  113260
                                       ;ITEM 33
 002236  073156  110716  114206                  .WORD     EM33,DH33,DT33,DF33
 002244  113263
                                       ;ITEM 34
 002246  073315  111145  114222                  .WORD     EM34,DH34,DT34,DF34
 002254  113301
                                       ;ITEM 35
 002256  073421  111145  114222                  .WORD     EM35,DH35,DT35,DF35
 002264  113301
                                       ;ITEM 36
 002266  073530  111225  114236                  .WORD     EM36,DH36,DT36,DF36
 002274  113306
                                       ;ITEM 37
 002276  073662  111272  114250                  .WORD     EM37,DH37,DT37,DF37
```

```
002304  113312
                                ;ITEM 40
002306  073744  111424  114276             .WORD   EM40,DH40,DT40,DF40
002314  113324
                                ;ITEM 41
002316  074117  111357  114264             .WORD   EM41,DH41,DT41,DF41
002324  113320
                                ;ITEM 42
002326  074303  111357  114264             .WORD   EM42,DH42,DT42,DF42
002334  113320
                                ;ITEM 43
002336  074556  111424  114276             .WORD   EM43,DH43,DT43,DF43
002344  113324
                                ;ITEM 44
002346  074704  111473  114320             .WORD   EM44,DH44,DT44,DF44
002354  113334
                                ;ITEM 45
002356  075100  111473  114320             .WORD   EM45,DH45,DT45,DF45
002364  113334
                                ;ITEM 46
002366  075277  111566  114360             .WORD   EM46,DH46,DT46,DF46
002374  113353
                                ;ITEM 47
002376  075420  111566  114360             .WORD   EM47,DH47,DT47,DF47
002404  113353
                                ;ITEM 50
002406  074704  111473  114412             .WORD   EM50,DH50,DT50,DF50
002414  113334
                                ;ITEM 51
002416  075100  111473  114412             .WORD   EM51,DH51,DT51,DF51
002424  113334
                                ;ITEM 52
002426  075277  111566  114452             .WORD   EM52,DH52,DT52,DF52
002434  113353
                                ;ITEM 53
002436  075420  111566  114452             .WORD   EM53,DH53,DT53,DF53
002444  113353
                                ;ITEM 54
002446  075544  111612  114504             .WORD   EM54,DH54,DT54,DF54
002454  113367

                                ;ITEM    55

002456  075717                             EM55
002460  111653                             DH55
002462  114514                             DT55
002464  000000                             0



                                ;ITEM    56

002466  075750                             EM56
002470  111653                             DH55
002472  114514                             DT55
002474  000000                             0
```

```


                                 ;ITEM    57

 002476  076016                           EM57
 002500  111653                           DH55
 002502  114514                           DT55
 002504  113372                           DF57


                                 ;ITEM    60

 002506  076057                           EM60
 002510  111653                           DH55
 002512  114514                           DT55
 002514  113372                           DF61

                                 ;ITEM    61

 002516  076116                           EM61
 002520  111653                           DH55
 002522  114514                           DT55
 002524  113372                           DF61


                                 ;ITEM    62

 002526  076205                           EM62
 002530  111653                           DH55
 002532  114514                           DT55
 002534  113372                           DF62


                                 ;ITEM    63

 002536  076237                           EM63
 002540  111653                           DH55
 002542  114514                           DT55
 002544  113372                           DF63


                                 ;ITEM    64

 002546  076315                           EM64
 002550  111653                           DH55
 002552  114514                           DT55
 002554  113372                           DF64


                                 ;ITEM    65

 002556  076376                           EM65
 002560  111653                           DH55
 002562  114514                           DT55
 002564  113372                           DF65

```

```

                                  ;ITEM   66

002566  076464                            EM66
002570  111667                            DH66
002572  114522                            DT66
002574  113374                            DF66


                                  ;ITEM   67

002576  076574                            EM67
002600  111667                            DH66
002602  114522                            DT66
002604  113374                            DF67


                                  ;ITEM   70

002606  076707                            EM70
002610  111667                            DH66
002612  114522                            DT66
002614  113374                            DF70

                                  ;ITEM   71

002616  077023                            EM71
002620  111733                            DH71
002622  114522                            DT66
002624  113374                            DF71

                                  ;ITEM   72

002626  077071                            EM72
002630  111667                            DH66
002632  114522                            DT66
002634  113374                            DF72

                                  ;ITEM   73

002636  077220                            EM73
002640  111667                            DH66
002642  114522                            DT66
002644  113374                            DF73

                                  ;ITEM   74

002646  077330                            EM74
002650  111667                            DH66
002652  114522                            DT66
002654  113374                            DF74

                                  ;ITEM   75

002656  077432                            EM75
002660  111667                            DH66
```

```
002662  114522                                  DT66
002664  113374                                  DF75

                                       ;ITEM    76

002666  077546                                  EM76
002670  111667                                  DH66
002672  114522                                  DT66
002674  113374                                  DF76

                                       ;ITEM    77

002676  077657                                  EM77
002700  111667                                  DH66
002702  114522                                  DT66
002704  113374                                  DF77

                                       ;ITEM    0

002706  000000  000000  000000                  .WORD   0,0,0,0
002714  000000

                                       ;ITEM    0

002716  000000  000000  000000                  .WORD   0,0,0,0
002724  000000

                                       ;ITEM    0

002726  000000  000000  000000                  .WORD   0,0,0,0
002734  000000

                                       ;ITEM    103
002736  100122                          EM103   ;NO PARITY ERROR TRAP ON VALID STORE PARITY ERROR
002740  111653                          DH55
002742  114514                          DT55
002744  113372                          DF103

                                       ;ITEM    104

002746  100203                          EM104   ;TEST-DATA-REFERENCE GIVING VALID STORE PARITY
002750  111653                          DH55    ;ERROR WAS NOT A MISS
002752  114514                          DT55
002754  113372                          DF104

                                       ;ITEM    105

002756  100307                          EM105   ;FVPE DID NOT GET CLEARED AFTER VSPE OCCURED
002760  111653                          DH55
002762  114514                          DT55
002764  113372                          DF105

                                       ;ITEM    106

002766  100363                          EM106   ;VALID-STORE-PARITY-ERROR BIT DID NOT SET IN CCR ONVSPE
002770  111653                          DH55
```

```
 002772  114514                       DT55
 002774  113372                       DF106

                                      ;ITEM   107

 002776  100453                       EM107   ;FAST ADDRESS MEMORY PARITY ERROR BITS (4,S) NOT
 003000  111765                       DH107   ;SET CORRECTLY IN MSER ON VSPE
 003002  114552                       DT107
 003004  113406                       DF107

                                      ;ITEM   110

 003006  100572                       EM110   ;VSIV SWITCHED ON VSPE
 003010  111653                       DH55
 003012  114514                       DT55
 003014  113372                       DF110

                                      ;ITEM   111

 003016  100620                       EM111   ;MEMORY SYSTEM ERROR REGISTER COULD NOT BE CLEARED
 003020  112037                       DH111
 003022  114514                       DT55
 003024  113372                       DF111

                                      ;ITEM   112

 003026  100702                       EM112   ;VSPE COULD NOT BE CLEARED IN CCR
 003030  111653                       DH55
 003032  114514                       DT55
 003034  113372                       DF112

                                      ;ITEM   113

 003036  100743                       EM113   ;TEST-DATA-REFERENCE NOT A HIT
 003040  111667                       DH66
 003042  114522                       DT66
 003044  113374                       DF113

                                      ;ITEM   0

 003046  000000  000000  000000               .WORD   0,0,0,0
 003054  000000

                                      ;ITEM   115

 003056  101001                       EM115   ;TEST DATA REFERENCE NOT A MISS
 003060  112054                       DH115   ;CACHE DID NOT TURN OFF ON BACK-TO-BACK FLUSH
 003062  114514                       DT55
 003064  113372                       DF115

                                      ;ITEM   116

 003066  000000  000000  000000               .WORD   0,0,0,0
 003074  000000

                                      ;ITEM   117
```

```

003076  000000  000000  000000           .WORD   0,0,0,0
003104  000000

                               ;ITEM   120

003106  000000  000000  000000           .WORD   0,0,0,0
003114  000000

                               ;ITEM   121

003116  000000  000000  000000           .WORD   0,0,0,0
003124  000000

                               ;ITEM   122

003126  000000  000000  000000           .WORD   0,0,0,0
003134  000000

                               ;ITEM 123
003136  101116                 EM123                       ;BYP BIT IN KIPDR COULD NOT BE CLEARED
003140  112073                 DH123                       ;    PC    KIPDR   (KIPDR)
003142  114566                 DT123                       ;$ERRPC,$REG0,$REG1,0
003144  113413                 DF123                       ;0,0,0
                               ;ITEM 124
003146  101164                 EM124                       ;BYP BIT IN KIPDR COULD NOT BE SET
003150  112073                 DH123
003152  114566                 DT123
003154  113413                 DF123

                               ;ITEM 125
003156  101226                 EM125                       ;TEST DATA COULD NOT BE MADE HIT
003160  112122                 DH125                       ; PC    CCR   PARADR  PAR  PDR   TST-DATA-ADR
003162  114576                 DT125                       ;$ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,0
003164  113400                 DF100

                               ;ITEM 126
003166  101266                 EM126                       ;TEST DATA REFERENCE NOT A MISS
                                                           ;CACHED DATA WAS NOT FORCED A MISS ON VIRTUAL PAGE BYPASS
003170  112122                 DH125
003172  114576                 DT125
003174  113400                 DF100

                               ;ITEM 127
003176  101417                 EM127                       ;TEST DATA REFERENCE NOT A MISS
                                                           ;CACHED DATA WAS NOT INVALIDATED ON VIRTUAL BYPASS
003200  112122                 DH125
003202  114576                 DT125
003204  113400                 DF100
                               ;ITEM 130
003206  101546                 EM130                       ;BYP BIT IN SIPDR COULD NOT BE CLEARED
003210  112211                 DH130                       ;    PC    SIPDR    (SIPDR)
003212  114566                 DT123
003214  113413                 DF123

                               ;ITEM 131
```

```
003216  101614                  EM131                              ;BYP IN SIPDR COULD NOT BE SET
003220  112211                  DH130
003222  114566                  DT123
003224  113413                  DF123

                                ;ITEM 132
003226  101656                  EM132                              ;BYP BIT IN UIPDR COULD NOT BE CLEARED
003230  112240                  DH132                              ;    PC       UIPDR     (UIPDR)
003232  114566                  DT123
003234  113413                  DF123

                                ;ITEM 133
003236  101724                  EM133                              ;BYP BIT IN UIPDR COULD NOT BE SET
003240  112240                  DH132
003242  114566                  DT123
003244  113413                  DF123
                                ;ITEM 0
003246  000000  000000  000000          .WORD    0,0,0,0
003254  000000
                                ;ITEM 0
003256  000000  000000  000000          .WORD    0,0,0,0
003264  000000
                                ;ITEM 136
003266  101766  112267  114614          .WORD    EM136,DH136,DT136,DF136
003274  113416
                                ;ITEM 137
003276  102203  112267  114614          .WORD    EM137,DH137,DT137,DF137
003304  113416
                                ;ITEM 140
003306  102421  112334  114626          .WORD    EM140,DH140,DT140,DF140
003314  113422
                                ;ITEM 141
003316  102762  112334  114626          .WORD    EM141,DH141,DT141,DF141
003324  113422
                                ;ITEM 142
003326  103322  112334  114626          .WORD    EM142,DH142,DT142,DF142
003334  113422
                                ;ITEM 143
003336  103664  112334  114626          .WORD    EM143,DH143,DT143,DF143
003344  113422
                                ;ITEM 144
003346  104225  112334  114626          .WORD    EM144,DH144,DT144,DF144
003354  113422
                                ;ITEM 145
003356  104557  112334  114626          .WORD    EM145,DH145,DT145,DF145
003364  113422
                                ;ITEM 146
003366  105110  112334  114626          .WORD    EM146,DH146,DT146,DF146
003374  113422
                                ;ITEM 147
003376  105443  112334  114626          .WORD    EM147,DH147,DT147,DF147
003404  113422
                                ;ITEM 150
003406  105775  112377  114640          .WORD    EM150,DH150,DT150,DF150
003414  113426
                                ;ITEM 151
```

```
003416  106061  112463  114652                  .WORD   EM151,DH151,DT151,DF151
003424  113432
                                        ;ITEM 152
003426  106061  112532  114652                  .WORD   EM152,DH152,DT152,DF152
003434  113432
                                        ;ITEM 153
003436  106061  112601  114652                  .WORD   EM153,DH153,DT153,DF153
003444  113432
                                        ;ITEM 154
003446  106142  112663  114662                  .WORD   EM154,DH154,DT154,DF154
003454  113435
                                        ;ITEM 155
003456  106174  112721  114662                  .WORD   EM155,DH155,DT155,DF155
003464  113435
                                        ;ITEM 156
003466  106226  112757  114662                  .WORD   EM156,DH156,DT156,DF156
003474  113435
                                        ;ITEM 0
003476  000000  000000  000000                  .WORD   0,0,0,0
003504  000000
                                        ;ITEM 160
003506  106273  113015  114652                  .WORD   EM160,DH160,DT160,DF160
003514  113435
                                        ;ITEM 161
003516  106325  113043  114652                  .WORD   EM161,DH161,DT161,DF161
003524  113435

                                        ;ITEM 162
003526  106373  113073  114674                  .WORD   EM162,DH162,DT162,DF55
003534  113372

                                        ;ITEM 163
003536  106603  113073  114674                  .WORD   EM163,DH162,DT162,DF55
003544  113372

                                        ;ITEM 164
003546  106675  113073  114674                  .WORD   EM164,DH162,DT162,DF55
003554  113372

                                        ;ITEM 165
003556  106754  113073  114674                  .WORD   EM165,DH162,DT162,DF55
003564  113372

                                        ;ITEM 166
003566  106773  113073  114674                  .WORD   EM166,DH162,DT162,DF55
003574  113372

                                        ;ITEM 167
003576  107057  113073  114674                  .WORD   EM167,DH162,DT162,DF55
003604  113372

                                        ;ITEM 170
003606  107163  113073  114674                  .WORD   EM170,DH162,DT162,DF55
003614  113372

                                        ;ITEM 171
```

```
 003616  107226  113073  114674                  .WORD   EM171,DH162,DT162,DF55
 003624  113372

                                         ;ITEM 172
 003626  107272  113073  114674                  .WORD   EM172,DH162,DT162,DF55
 003634  113372

                                         ;ITEM 173
 003636  107356  113073  114674                  .WORD   EM173,DH162,DT162,DF55
 003644  113372

                                         ;ITEM 174
 003646  107544  113073  114674                  .WORD   EM174,DH162,DT162,DF55
 003654  113372

                                         ;ITEM 175
 003656  107607  113073  114674                  .WORD   EM175,DH162,DT162,DF55
 003664  113372
 003666  000016                          RS4REG: .WORD   16
 003670  172040                          RS4CS1: .WORD   172040
 003672  000000                          RS4WC:  .WORD   0
 003674  000000                          RS4BA:  .WORD   0
 003676  000000                          RS4DA:  .WORD   0
 003700  000000                          RS4CS2: .WORD   0
 003702  000000                          RS4DS:  .WORD   0
 003704  000000                          RS4ER:  .WORD   0
 003706  000000                          RS4AS:  .WORD   0
 003710  000000                          RS4LA:  .WORD   0
 003712  000000                          RS4DB:  .WORD   0
 003714  000000                          RS4MR:  .WORD   0
 003716  000000                          RS4DT:  .WORD   0
 003720  000000                          RS4BAE: .WORD   0
 003722  000000                          RS4CS3: .WORD   0

 003724  000026                          RP4REG: .WORD   26
 003726  176700                          RP4CS1: .WORD   176700
 003730  000000                          RP4WC:  .WORD   0
 003732  000000                          RP4BA:  .WORD   0
 003734  000000                          RP4DA:  .WORD   0
 003736  000000                          RP4CS2: .WORD   0
 003740  000000                          RP4DS:  .WORD   0
 003742  000000                          RP4RR1: .WORD   0
 003744  000000                          RP4AS:  .WORD   0
 003746  000000                          RP4LA:  .WORD   0
 003750  000000                          RP4DB:  .WORD   0
 003752  000000                          RP4MR:  .WORD   0
 003754  000000                          RP4DT:  .WORD   0
 003756  000000                          RP4SN:  .WORD   0
 003760  000000                          RP4OF:  .WORD   0
 003762  000000                          RP4DC:  .WORD   0
 003764  000000                          RP4CCC: .WORD   0
 003766  000000                          RP4RR2: .WORD   0
 003770  000000                          RP4RR3: .WORD   0
 003772  000000                          RP4EC1: .WORD   0
 003774  000000                          RP4EC2: .WORD   0
 003776  000000                          RP4BAE: .WORD   0
```

```
004000 000000                      RP4CS3: .WORD   0

004002 000014                      RH4REG: .WORD   14
004004 160100                      RH4CS1: .WORD   160100
004006 000000                      RH4WC:  .WORD   0
004010 000000                      RH4BA:  .WORD   0
004012 000000                      RH4MR2: .WORD   0
004014 000000                      RH4CS2: .WORD   0
004016 000000                      RH4ST:  .WORD   0
004020 000000                      RH4ER:  .WORD   0
004022 000000                      RH4AS:  .WORD   0
004024 000000                      RH4DR:  .WORD   0
004026 000000                      RH4DB:  .WORD   0
004030 000000                      RH4MR1: .WORD   0
004032 000000                      RH4DT:  .WORD   0

004034 000002                      RH4REX: .WORD   2
004036 160174                      RH4AE:  .WORD   160174
004040 000000                      RH4CS3: .WORD   0

004042 000007                      RK5REG: .WORD   7
004044 177400                      RK5DS:  .WORD   177400
004046 000000                      RK5ER:  .WORD   0
004050 000000                      RK5CS1: .WORD   0
004052 000000                      RK5WC:  .WORD   0
004054 000000                      RK5BA:  .WORD   0
004056 000000                      RK5DA:  .WORD   0
004060 000000                      RK5DB:  .WORD   0


004062 000006                      UBEREG: .WORD   6
004064 170000                      UBEDB:  .WORD   170000
004066 000000                      UBECC:  .WORD   0
004070 000000                      UBEBA:  .WORD   0
004072 000000                      UBECR1: .WORD   0
004074 000000                      UBECLR: .WORD   0
004076 000000                      UBECR2: .WORD   0

                                   ;THESE ARE THE DEVICE TRAP VECTER ADDRESSES:
004100 000204                      RS4V:   .WORD   204
004102 000254                      RP4V:   .WORD   254
004104 000774                      RH4V:   .WORD   774
004106 000220                      RK5V:   .WORD   220
004110 000510                      UBEV:   .WORD   510



004112 005037 001502               START:  CLR     $TSTNM
004116 012737 000340 177776                MOV     #340,@#PS          ;;LOCK OUT ALL INTERRUPTS
004124 012706 001500                       MOV     #$CMTAG,R6         ;;FIRST LOCATION TO BE CLEARED
004130 005026                              CLR     (R6)+              ;;CLEAR MEMORY LOCATION
004132 022706 001540                       CMP     #$TKS,R6           ;;DONE?
004136 001374                              BNE     .-6                ;;LOOP BACK IF NO
004140 012706 001500                       MOV     #STACK,SP          ;;SETUP THE STACK POINTER
004144 012737 051354 000020                MOV     #$SCOPE,@#IOTVEC   ;;IOT VECTOR FOR SCOPE ROUTINE
004152 012737 000340 000022                MOV     #340,@#IOTVEC+2    ;;LEVEL 7
```

```
004160  012737  051636  000030            MOV     #$ERROR,@#EMTVEC  ;;EMT VECTOR FOR ERROR ROUTINE
004166  012737  000340  000032            MOV     #340,@#EMTVEC+2   ;;LEVEL 7
004174  012737  053112  000034            MOV     #$TRAP,@#TRAPVEC  ;;TRAP VECTOR FOR TRAP CALLS
004202  012737  000340  000036            MOV     #340,@#TRAPVEC+2;LEVEL 7
004210  012737  053206  000024            MOV     #$PWRDN,@#PWRVEC  ;;POWER FAILURE VECTOR
004216  012737  000340  000026            MOV     #340,@#PWRVEC+2   ;;LEVEL 7
004224  013737  051250  051242            MOV     $ENDCT,$EOPCT     ;;SETUP END-OF-PROGRAM COUNTER
004232  005037  001676                    CLR     $TIMES            ;;INITIALIZE NUMBER OF ITERATIONS
004236  005037  001700                    CLR     $ESCAPE           ;;CLEAR THE ESCAPE ON ERROR ADDRESS
004242  112737  000001  001517            MOVB    #1,$ERMAX         ;;ALLOW ONE ERROR PER TEST
004250  012737  004250  001510            MOV     #.,$LPADR         ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
004256  012737  004256  001512            MOV     #.,$LPERR         ;;SETUP THE ERROR LOOP ADDRESS
004264  005227  177777                    INC     #-1               ;;FIRST TIME?
004270  001044                            BNE     64$               ;;BRANCH IF NO
004272  022737  051320  000042            CMP     #$ENDAD,@#42      ;;ACT-11?
004300  001440                            BEQ     64$               ;;BRANCH IF YES
004302  104400  004310                    TYPE    ,65$              ;;TYPE ASCIZ STRING
004306  000435                            BR      64$               ;;GET OVER THE ASCIZ
                                  ;;65$:   .ASCIZ  <CRLF>'CEKBD-D  PDP 11/70-74MP CACHE MEMORY DIAGNOSTIC PART 2 '<CRLF>
004402                            64$:
004402  005227  177777                    INC     #-1               ;;FIRST TIME?
004406  001043                            BNE     66$               ;;BRANCH IF NO
004410  022737  051320  000042            CMP     #$ENDAD,@#42      ;;ACT-11?
004416  001437                            BEQ     66$               ;;BRANCH IF YES
004420  104400  004426                    TYPE    ,67$              ;;TYPE ASCIZ STRING
004424  000434                            BR      66$               ;;GET OVER THE ASCIZ
                                  ;;67$:   .ASCIZ  <CRLF>'PROGRAMMABLE RP04 DRIVES WILL NOT BE USED BY TEST 35'<CRLF>
004516                            66$:
                                  ;;***********************************************************


                                  ; SIZE MEMORY AND COMPARE IT WITH THE SYSTEM SIZE REGISTER
                                  ;PRINT A WARNING MESSAGE IF THEY DISAGREE.

004516  052737  000200  053410            BIS     #BIT07,$KT11
004524  004737  053342                    JSR     PC,$SIZE
004530  062737  000037  053726            ADD     #37,$LSTBK                ;ADJUST THE SIZE FOR COMPARISON
                                                                            ;TO SIZE REGISTER
004536  023737  177760  053726            CMP     @#SIZELO,$LSTBK           ;IS THE ACTUAL SIZE REFLECTED BY THE
                                                                            ;SIZE REGISTER?
004544  001420                            BEQ     OKSIZ
004546  104400  070007                    TYPE    ,MS01
004552  104400  070144                    TYPE    ,MS02
004556  104400  001707                    TYPE    ,$CRLF
004562  013746  177760                    MOV     @#SIZELO,-(SP)    ;;SAVE @#SIZELO FOR TYPEOUT
004566  104404                            TYPOS                     ;;GO TYPE--OCTAL ASCII
004570     006                            .BYTE   6                 ;;TYPE 6 DIGIT(S)
004571     000                            .BYTE   0                 ;;SUPPRESS LEADING ZEROS
004572  104400  070173                    TYPE    ,MS03
004576  013746  053726                    MOV     $LSTBK,-(SP)      ;;SAVE $LSTBK FOR TYPEOUT
004602  104404                            TYPOS                     ;;GO TYPE--OCTAL ASCII
004604     006                            .BYTE   6                 ;;TYPE 6 DIGIT(S)
004605     000                            .BYTE   0                 ;;SUPPRESS LEADING ZEROS
004606                            OKSIZ:

                                  ;;
                                  ;;*** TEST FOR VARIOUS KB11 PROCESSORS ***
```

```
                                        ;;
                                        ;;*THIS ROUTINE POLES THE RESULTS OF ATTEMPTS TO SET TO ONE
                                        ;;*CERTAIN CRITICAL BITS THAT ARE KNOWN TO BE OPERATIVE ON A KB11CM,
                                        ;;*OR KB11EM PROCESSOR. IF TWO OUT OF FOUR OF THE TESTS ARE
                                        ;;*POSITIVE THEN THE KB11CM OR KB11EM FLAG IS SET,IF LESS THAN TWO OF THE
                                        ;;*TESTS ARE POSITIVE THEN THE KB11E FLAG OR NO FLAG IS SET. THE DETERMINATION
                                        ;;*OF WHICH PAIR IS VALID IS BASED ON THE RESULTS OF EXECUTING AN MFPT OPCODE
                                        ;;*(OPCODE 7).  IF THIS INSTRUCTION TRAPS THIS IS AN KB11CM OR
                                        ;;*A PLAIN 1170 (KB11-B OR KB11-C).  IF THE INSTRUCTION DOES NOT TRAP THEN
                                        ;;*THIS IS A KB11-E OR KB11-EM.
                                        ;
004606  105037  001714                  KBTST:  CLRB    @#KB11CM            ;RESET THE MP FLAG
004612  005037  001712                          CLR     @#KB11E             ;CLEAR KB11E AND KB11EM FLAGS
004616  012737  005054  000010                  MOV     #MFPTTR,@#RESVEC    ;SET UP TRAP ADDRESS FOR MFPT AT RESERV VECTOR
004624  000007                                  MFPT                        ;EXECUTE MFPT. WILL TRAP ON 1170 (KB11B/C) OR
                                                                            ;KB11CM (11/74                  )
004626  012737  000001  001712                  MOV     #1,@#KB11E          ;HERE IF KB11E OR KB11EM. SET FLAG
004634  005037  177750                  T1:     CLR     @#MAINT             ;CLEAR THE MAINTENANCE REGISTER
004640  005005                                  CLR     R5                  ;RESET THE TEST COUNTER
004642  012700  177746                          MOV     #CONTRL,R0          ;GET THE ADDRESS OF...
004646  012701  177750                          MOV     #MAINT,R1           ;CCR,MAINT,AND MAPH00...
004652  012702  170202                          MOV     #MAPH00,R2          ;AND PLACE IN R0-R2
004656  052710  040000                          BIS     #BIT14,(R0)         ;TRY TO SET IVSS BIT
004662  032710  040000                          BIT     #BIT14,(R0)         ;DID IT SET?
004666  001403                                  BEQ     T2                  ;NO,GO TO NEXT TEST
004670  042710  040000                          BIC     #BIT14,(R0)         ;CLEAR IT.
004674  005205                                  INC     R5                  ;TEST IS POSITIVE
004676  052711  000001                  T2:     BIS     #BIT0,(R1)          ;SET EDMA IN MAINT REGISTER
004702  032711  000001                          BIT     #BIT0,(R1)
004706  001410                                  BEQ     T3
004710  052710  004000                          BIS     #BIT11,(R0)         ;TRY TO SET DMMA IN CCR
004714  032710  004000                          BIT     #BIT11,(R0)
004720  001403                                  BEQ     T3
004722  042710  004000                          BIC     #BIT11,(R0)
004726  005205                                  INC     R5
004730  042711  000001                  T3:     BIC     #BIT0,(R1)          ;MAKE SURE EDMA IS CLEAR
004734  052737  100000  172300                  BIS     #BIT15,KIPDR0       ;TRY TO SET BYP ON A PDR
004742  032737  100000  172300                  BIT     #BIT15,KIPDR0
004750  001404                                  BEQ     T4
004752  042737  100000  172300                  BIC     #BIT15,KIPDR0
004760  005205                                  INC     R5
004762  052712  100000                  T4:     BIS     #BIT15,(R2)         ;TRY TO SET BYP ON UNIBUS MAP
004766  032712  100000                          BIT     #BIT15,(R2)
004772  001403                                  BEQ     T.END
004774  042712  100000                          BIC     #BIT15,(R2)
005000  005205                                  INC     R5
005002  022705  000002                  T.END:  CMP     #2,R5               ;IS THE RESULT OF THE TEST >=2
005006  101021                                  BHI     2$                  ;NO,THIS IT A KB11E OR KB11-B/C (11/70)
005010  005000                                  CLR     R0
005012  005037  177746                          CLR     @#CONTRL            ;CLEAR CACHE CONT. REG. AND
005016  013701  177746                  3$:     MOV     @#CONTRL,R1         ;WAIT UNTILL VCIP BIT CLEARS
005022  001402                                  BEQ     4$                  ;OR THE COUNT RUNS OUT
005024  005200                                  INC     R0
005026  001373                                  BNE     3$
005030  005737  001712                  4$:     TST     @#KB11E             ;IS IS A KB11-E OR KB11-EM?
005034  001404                                  BEQ     1$                  ;BR IF NEITHER. MUST BE KB11CM
```

```
005036  012737  000400  001712                  MOV     #BIT8,@#KB11E    ;SET UPPER BYTE (KB11-EM)
005044  000402                                  BR      2$               ;DONE
005046  105237  001714                  1$:     INCB    @#KB11CM         ;YES, FLAG THIS AS A MODIFIED PROCESSOR
005052  000403                          2$:     BR      ENDKB            ;DONE DETERMINING WHICH CPU

005054                                  MFPTTR:                          ;HERE IF MFPT TRAPPED. SEE IF 1170 OR KB11CM
005054  012716  004634                          MOV     #T1,(SP)         ;SET UP RETURN ADDRESS FOR RTI
005060  000002                                  RTI                      ;RETURN
005062                                  ENDKB:
005062  005227  177777                          INC     #-1              ;FIRST TIME?
005066  001026                                  BNE     100$             ;BR IF NO
005070  104400  070203                          TYPE    ,MSG1            ;<15><12>CPU UNDER TEST FOUND TO BE A
005074  005737  001712                          TST     @#KB11E          ;IS THIS A KB11-E OR KB11-EM?
005100  001011                                  BNE     101$             ;BR IF EITHER ONE
005102  105737  001714                          TSTB    @#KB11CM         ;IS IT A 11/74          (KB11CM)
005106  001003                                  BNE     1$               ;BR IF IT IS
005110  104400  070253                          TYPE    ,MSG3            ;KB11-B/C<15><12>
005114  000413                                  BR      100$             ;SKIP OTHER MESSAGE
005116  104400  070265                  1$:     TYPE    ,MSG4            ;11/74          (KB11CM)<15><12>
005122  000410                                  BR      100$             ;SKIP CISP MESSAGE
005124  105737  001712                  101$:   TSTB    @#KB11E          ;IS IT A KB11-E?
005130  001403                                  BEQ     102$             ;BR IF NOT. MUST BE KB11-EM
005132  104400  070316                          TYPE    ,MSG5            ;KB11-E<15><12>
005136  000402                                  BR      100$             ;SKIP KB11-EM MESSAGE
005140  104400  070242                  102$:   TYPE    ,MSG2            ;KB11-EM<15><12>
005144                                  100$:

                                        ;THIS ROUTINE SAVES THE TOP 1500 (DEC) WORDS OF THE FIRST 28K OF
                                        ;MEMORY. THESE LOCATIONS SHOULD CONTAIN EITHER THE MONITOR OR THE
                                        ;LOADER WHICH LOADED THE PROGRAM. NOTE THAT TO RESTORE THIS PART
                                        ;OF CORE, THAT IS TO RESTORE THE LOADER OR MONITOR, ALL THE USER
                                        ;MUST DO IS TYPE ^C (CONTROL-C), WHILE THIS PROGRAM IS RUNNING.
                                        ;THIS WILL AUTOMATICALLY RESTORE THE TOP PART OF MEMORY TO ITS STATE
                                        ;BEFORE THIS PROGRAM WAS STARTED! AFTER THE MONITOR (OR LOADER) HAS BEEN
                                        ;RESTORED THIS PROGRAM WILL HALT.
005144  005237  054644                  LOOP:   INC     MONF             ;INCREMENT THE FLAG WHICH INDICATES
005150  001013                                  BNE     TOP              ;WHETHER OR NOT THE TOP OF MEMORY
                                                                         ;IN THE FIRST 28K HAS BEEN SAVED.
005152  013737  000060  054642                  MOV     @#TKVEC,MONTTY   ;SAVE THE INITIAL CONTENTS OF THE TTY
                                                                         ;KEYBOARD INTERRUPT VECTOR.
005160  012700  002734                          MOV     #^D1500,R0       ;IF NOT THEN SAVE IT.
005164  012701  116710                          MOV     #BOTTOM+4,R1     ;SAVE IT AT THE BOTTOM OF THIS PROGRAM.
005170  012702  160000                          MOV     #160000,R2       ;GET THE ADDRESS OF THE END OF THE MONITOR.
005174  014221                          1$:     MOV     -(R2),(R1)+      ;SAVE 1500 (DEC) LOCATIONS (WORDS)
005176  077002                                  SOB     R0,1$
005200  012737  000044  177770          TOP:    MOV     #44,@#177770

005206  012737  054512  000060                  MOV     #RESMON,@#TKVEC ;SET THE KEYBOARD INTERRUPT VECTOR.
005214  012737  000340  000062                  MOV     #340,@#TKVEC+2
005222  005077  174314                          CLR     @$TKB            ;MAKE SURE THE KEYBOARD BUFFER IS CLEAR.
005226  152777  000100  174304                  BISB    #BIT6,@$TKS      ;TURN ON INTERRUPT ENABLE FOR THE KEYBOARD.
005234  012737  054050  000004                  MOV     #CPSPUR,@#4      ;SET UP FOR UNEXPECTED ERRORS.
005242  012737  054076  000114                  MOV     #SPUR,@#114


                                        ;;*******************************************************************
                                        ;*TEST 1         PARITY ERROR ABORT
```

```
                                ;*
                                ;*      THIS TEST ENSURES THAT A CACHE PARITY ERROR FLAG CAUSES AN ABORT.
                                ;*      THIS IS DONE BY FORCING A PARITY ERROR ON AN EVEN WORD.
                                ;;*************************************************************************
005250  000004                  TST1:   SCOPE
005252  012737  005424  001700          MOV     #2$,$ESCAPE       ;SETUP ESCAPE ADDRESS
005260  012737  000014  177746          MOV     #14,@#CONTRL      ;ENSURE MISSES TO BOTH GROUPS
005266  012737  005334  001512          MOV     #7$,$LPERR        ;SETUP ERROR LOOP
005274  012737  005424  000114          MOV     #2$,@#CACHVEC     ;SETUP CACHE VECTOR
005302  012737  005360  000004          MOV     #3$,@#ERRVEC      ;SETUP LOCATION 4
005310  012737  005374  000014          MOV     #4$,@#14          ;SETUP LOCATION 14
005316  012737  005410  000104          MOV     #5$,@#104         ;SETUP LOCATION 104
005324  012704  170000                  MOV     #170000,R4        ;PUT MAINTENANCE DATA IN R4
005330  012702  177750                  MOV     #MAINT,R2         ;PUT ADDRESS OF MAIN REG IN R2
005334  012706  001500          7$:     MOV     #STACK,SP         ;INITIALIZE THE SP
005340  000401                          BR      1$                ;GO TO NEXT INSTRUCTION
        005342                          LOC=.                     ;THIS IS
        005340                          LOC=-3&LOC                ;USED TO MAKE
        005344                          LOC=LOC+4                 ;1$ FALL ON
        005344                          .=LOC                     ;AND EVEN WORD
005344  000240                  1$:     NOP                       ;USED TO MAKE BAD PARITY INSTR ON EVEN WORD
005346  010412                          MOV     R4,(R2)           ;SET BITS IN MAINT REG
005350  005701                          TST     R1                ;EXECUTE INSTR TO CAUSE PE ABORT
                                ;FAILURE, NO ABORT
005352  005012                          CLR     (R2)              ;CLEAR MAINT REG
005354  000240                          NOP
005356  104162                          ERROR   162               ;NO PE ABORT
                                ;FAILURE, ABORTED TO WRONG VECTOR
005360  005012                  3$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005362  000240                          NOP
005364  012737  177777  177744          MOV     #-1,@#MEMERR
005372  104163                          ERROR   163               ;ABORTED TO LOCATION 4
005374  005012                  4$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005376  000240                          NOP
005400  012737  177777  177744          MOV     #-1,@#MEMERR
005406  104164                          ERROR   164               ;ABORTED TO 14
005410  005012                  5$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005412  000240                          NOP
005414  012737  177777  177744          MOV     #-1,@#MEMERR
005422  104165                          ERROR   165               ;ABORTED TO 104
                                ;TEST OK
005424  005012                  2$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005426  000240                          NOP
005430  012737  177777  177744          MOV     #-1,@#MEMERR      ;CLEAR MEMORY ERROR REG
005436  012737  054050  000004          MOV     #CPSPUR,@#ERRVEC  ;RESET LOCATION 4
                                                                  ;CONTINUE
                                ;;*************************************************************************
                                ;*TEST 2        PARITY ERROR TRAP
                                ;*
                                ;*      THIS TEST ENSURES THAT A PARITY TRAP FUNCTIONS PROPERLY.
                                ;*      THIS IS DONE BY MAKING THE ODD WORD HAVE BAD PARITY.
                                ;*      IF THE TRAP DOESN'T OCCUR THEN THE PROBLEM IS ON TMCA.
                                ;*      IF A TRAP OCCURS TO THE WRONG VECTOR THE PROBLEM COULD BE
                                ;*      ON TMCA OR UBCB.
                                ;;*************************************************************************
005444  000004                  TST2:   SCOPE
```

```
005446  012737  005552  001700            MOV     #3$,$ESCAPE       ;SETUP ESCAPE ADDRESS
005454  012737  005506  001512            MOV     #1$,$LPERR        ;SETUP ERROR LOOP
005462  012737  005536  000004            MOV     #2$,@#ERRVEC      ;SETUP THE ERROR VECTOR
005470  012737  005552  000114            MOV     #3$,@#CACHVEC     ;SETUP THE CACHE VECTOR
005476  012704  170000                    MOV     #170000,R4        ;PUT MAINT DATA IN R4
005502  012702  177750                    MOV     #MAINT,R2         ;PUT MAINT REG ADDR IN R2
005506  012706  001500            1$:     MOV     #STACK,SP         ;INITIALIZE THE SP
005512  000402                            BR      4$                ;GO TO NEXT INSTRUCTION
        005514                            LOC=.                     ;THIS IS USED
        005514                            LOC=-3&LOC                ;TO MAKE
        005520                            LOC=LOC+4                 ;1$ FALL ON
        005520                            .=LOC                     ;AN EVEN WORD
005520  000240                    4$:     NOP                       ;GOOD PARITY ON EVEN WORD
005522  010412                            MOV     R4,(R2)           ;SET BITS IN MAINT REG
005524  000240                            NOP
005526  005701                            TST     R1
                                  ;FAILURE, NO TRAP
005530  005012                            CLR     (R2)              ;ENSURE MAINT REG CLEAR
005532  000240                            NOP
005534  104166                            ERROR   166               ;NO PE TRAP
                                  ;FAILURE, TRAPPED TO WRONG VECTOR
005536  005012                    2$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005540  000240                            NOP
005542  012737  177777  177744            MOV     #-1,@#MEMERR      ;CLEAR MEM ERROR REG
005550  104167                            ERROR   167               ;PE TRAP, TRAPPED TO
                                  ;TEST OK
005552  005012                    3$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005554  000240                            NOP
005556  012737  177777  177744            MOV     #-1,@#MEMERR      ;CLEAR MEM ERROR REG
005564  012737  054050  000004            MOV     #CPSPUR,@#ERRVEC ;RESTORE LOCATION 4
005572  012737  054076  000250            MOV     #SPUR,@#MMVEC ;RESTORE MEM VEC
                                                                    ;CONTINUE
                                  ;;*****************************************************************
                                  ;*TEST 3          MEM MGT AND PE TRAP PRIORITY ARBITRATION
                                  ;*
                                  ;*       THIS TEST ENSURES THAT THE ARBITRATION LOGIC WORKS FOR MEMORY
                                  ;*       MANAGEMENT AND PARITY ERROR TRAPS.
                                  ;*
                                  ;;*****************************************************************
005600  000004                    TST3:   SCOPE
005602  012737  001400  172354    1$:     MOV     #1400,@#KIPAR6    ;RESTORE PAR6
005610  112737  000004  172314            MOVB    #4,@#KIPDR6       ;SETUP PAGE 6 TO TRAP ON ALL ACCESSES
005616  012704  170000                    MOV     #170000,R4        ;PUT MAINT REG DATA IN R4
005622  012702  177750                    MOV     #MAINT,R2         ;PUT ADDRESS OF MAINT REG IN R2
                                  ;;*****************************************************************
                                  ;PIR6 DISABLED BY MGMT
```

```
005626  012737  040000  140000                  MOV     #BIT14,@#140000 ;PUT PIR6 ENABLE BIT IN PAGE 6
005634  012737  005706  000240                  MOV     #3$,@#PIRQVEC   ;SETUP PIRQ VECTOR
005642  012737  000340  000252                  MOV     #PR7,@#MMVEC+2  ;SET UP MMVEC PSW
005650  012737  005720  000250                  MOV     #4$,@#MMVEC     ;SETUP MEM MGMT VECTOR
005656  012737  005664  001512                  MOV     #5$,$LPERR      ;SETUP ERROR LOOP
005664  012706  001500                  5$:     MOV     #STACK,SP       ;INITIALIZE THE SP
005670  012737  001001  177572                  MOV     #1001,@#MMR0    ;TURN RELOCATION ON
005676  000235                                  SPL     5               ;SET PROCESSOR AT LEVEL 5
005700  013737  140000  177772                  MOV     @#140000,@#PIRQ ;SET PIR6 AND MEM MGT TRAP
                                        ;FAILURE, PRI6 CAME THRU
005706  005037  177572                  3$:     CLR     @#MMR0          ;TURN RELOCATION OFF
005712  005037  177772                          CLR     @#PIRQ          ;CLEAR PIR6
005716  104170                                  ERROR   170     ;PIR6 CAME IN ON
                                        ;
                                        ;;*******************************************************************
                                        ;PIR3 DISABLED BY MGMT
005720  005037  177572                  4$:     CLR     @#MMR0          ;TURN RELOCATION OFF
005724  005037  177772                          CLR     @#PIRQ          ;CLEAR PIR LEVEL 6
005730  012737  006002  000240                  MOV     #6$,@#PIRQVEC   ;SETUP PIRQ VECTOR
005736  012737  006014  000250                  MOV     #7$,@#MMVEC     ;SETUP MEM MGT VECTOR
005744  012737  005760  001512                  MOV     #8$,$LPERR      ;SETUP ERROR LOOP
005752  012737  004000  140000                  MOV     #BIT11,@#140000 ;PUT PIR3 ENABLE BIT IN PAGE 6
005760  012706  001500                  8$:     MOV     #STACK,SP       ;INITIALIZE THE SP
005764  012737  001001  177572                  MOV     #1001,@#MMR0    ;TURN ON RELOCATION
005772  000232                                  SPL     2               ;LOWER CPU TO LEVEL 2
005774  013737  140000  177772                  MOV     @#140000,@#PIRQ ;SET PIR3 & MGMT
                                        ;FAILURE, PIR3 CAME THRU
006002  005037  177572                  6$:     CLR     @#MMR0          ;TURN OFF RELOCATION
006006  005037  177772                          CLR     @#PIRQ          ;CLEAR PIR3
006012  104171                                  ERROR   171             ;PIR3 CAME IN ON
                                        ;
                                        ;;*******************************************************************
                                        ;STACK LIMIT YELLOW DISABLED BY PARITY ERROR
006014  005037  177572                  7$:     CLR     @#MMR0          ;TURN RELOCATION OFF
006020  005037  177772                          CLR     @#PIRQ          ;CLEAR PIR LEVEL 3
006024  012737  006064  001512                  MOV     #9$,$LPERR      ;SETUP ERROR LOOP
006032  012737  006110  000004                  MOV     #10$,@#ERRVEC   ;SETUP THE ERROR VECTOR
006040  012737  006110  000114                  MOV     #10$,@#CACHVEC  ;SETUP CACHEVEC
006046  012737  000240  000116                  MOV     #PR5,@#CACHVEC+2 ;PUT PRIORITY 5 IN CACHE VECTOR PSW
006054  012704  170000                          MOV     #170000,R4      ;PUT MAINT REG DATA IN R4
006060  012702  177750                          MOV     #MAINT,R2       ;PUT ADDRESS OF MAINT ON R2
006064  005037  000370                  9$:     CLR     @#370           ;ENSURE LOCATION 370 CLEAR
006070  012706  000376                          MOV     #376,SP         ;SETUP THE SP TO YELLOW ZONE
006074  000401                                  BR      11$             ;GO TO 12$
        006076                                  LOC=.                   ;THIS MAKES
        006074                                  LOC=-3&LOC              ;THE NEXT INSTRUCTION
        006100                                  LOC=LOC+4               ;FALL ON
        006100                                  .=LOC                   ;AN EVEN WORD
006100  010412                          11$:    MOV     R4,(R2)         ;SET MAINT REG
006102  000240                                  NOP                     ;ODD WORD GOOD PARITY
006104  005216                                  INC     (SP)            ;CAUSE YEL ZONE (GOOD PARITY)
006106  005701                                  TST     R1              ;ODD WORD BAD PARITY
                                        ;SHOULD TAKE PE TRAP THEN YEL ZONE TRAP
006110  005012                          10$:    CLR     (R2)            ;CLEAR MAINTENANCE REGISTER
006112  000240                                  NOP
006114  022737  000240  000370                  CMP     #PR5,@#370      ;DID CACHVEC PSW GET STACKER?
```

```
006122  001403                          BEQ     12$                 ;BRANCH IF YES
006124  012706  001500                  MOV     #STACK,SP           ;RESTORE THE SP
006130  104172                          ERROR   172                 ;YEL ZONE CAME THRU ON PE TRAP
                                ;;*************************************************************
                                ;MEMORY MANAGEMENT TRAP DISABLED BY PARITY TRAP
006132  012737  006200  001512  12$:    MOV     #13$,$LPERR         ;SETUP ERROR LOOP
006140  012737  006230  000250          MOV     #15$,@#MMVEC        ;SETUP MEM MGT VECTOR
006146  012737  006230  000114          MOV     #15$,@#CACHVEC      ;SETUP CACHVEC
006154  012737  000340  000116          MOV     #PR7,@#CACHVEC+2    ;RESTORE EACH VEC PSW
0061 2  012704  170000                  MOV     #170000,R4          ;PUT MAINT DATA IN R4
006166  012702  177750                  MOV     #MAINT,R2           ;PUT ADDRESS OF MAINT REG IN R2
006172  112737  000004  172314          MOVB    #4,@#KIPDR6         ;ENSURE PAGE 6 TRAPS
006200  012706  001500          13$:    MOV     #STACK,SP           ;INITIALIZE THE SP
006204  012737  001001  177572          MOV     #1001,@#MMR0        ;TURN RELOCATION ON
006212  000402                          BR      16$
        006214                          LOC=.
        006214                          LOC=-3&LOC
        006220                          LOC=LOC+4
        006220                          .=LOC
006220  010412                  16$:    MOV     R4,(R2)             ;SET MAINT REG (PARITY GOOD)
006222  000240                          NOP                         ;ODD WORD PARITY GOOD
006224  005237  140402                  INC     @#140402            ;INC HAS GOOD PARITY BUT ADDRESS
                                                                    ;HAS BAD PARITY. CAUSES MM TRAP
                                                                    ;AND PE TRAP
                                ;TEST OK
006230  005012                  15$:    CLR     (R2)                ;CLEAR MAINT REG
006232  000240                          NOP
006234  005037  177572                  CLR     @#MMR0              ;TURN RELOCATION OFF
006240  026627  000002  000340          CMP     2(SP),#PR7          ;DID PE TRAP OCCUR FIRST?
006246  001401                          BEQ     14$                 ;BRANCH IF YES
006250  104173                          ERROR   173                 ;MEM MGT TRAP CAME
006252  012737  054050  000004  14$:    MOV     #CPSPUR,@#ERRVEC    ;RESTORE LOCATION 4
006260  012737  054076  000114          MOV     #SPUR,@#CACHVEC     ;RESTORE LOCATION 114
006266  012737  177777  177744          MOV     #-1,@#MEMERR        ;CLEAR MEM ERROR REG
006274  005037  177766                  CLR     @#CPUERR            ;ENSURE CPUERROR CLEAR
                                                                    ;CONTINUE
                                ;
                                ;;*************************************************************
                                ;*      THE NEXT TEST USES THE MAPPING BOX AND THE CACHE TO
                                ;*      GENERATE A PARITY ERROR ON THE UNIBUS.
                                ;;*************************************************************
                                ;*TEST 4        UNIBUS PARITY ERROR
                                ;*
                                ;*      THIS TEST MAKES A REFERENCE TO MEMORY THRU THE MAPPING
                                ;*      BOX THAT WILL CAUSE A PARITY ERROR.  IF THE ABORT DOESN'T
                                ;*      HAPPEN THEN THE PROBLEM IS ON UBCB.
                                ;*
                                ;*      NOTE:   MAP REGISTER 0 AND 1 ARE NOT USED IN CASE THE PROGRAM
                                ;*      IS RUNNING UNDER ACT11.
                                ;;*************************************************************
006300  000004                  TST4:   SCOPE
006302  012737  077406  172314          MOV     #77406,@#KIPDR6     ;SETUP PDR6
006310  012737  000060  172516          MOV     #60,@#MMR3          ;SETUP MMR3
006316  012706  001500                  MOV     #STACK,SP           ;INITIALIZE THE SP
006322  012700  170210                  MOV     #MAPL2,R0           ;GET ADDRESS OF MAP REG 2
006326  012701  000036                  MOV     #36,R1              ;SETUP SOB COUNT
```

```
006332  012737  006344  000004          MOV     #5$,@#ERRVEC      ;SETUP ERROR VECTOR
006340  005720                  8$:     TST     (R0)+             ;SEE IF MAP REG IS ENABLED
006342  000420                          BR      6$                ;BRANCH IF YES
006344  062700  000002          5$:     ADD     #2,R0             ;ADJUST R0 TO NEXT REGISTER
006350  077105                          SOB     R1,8$             ;TEST NEXT REGISTER
006352  012706  001500          7$:     MOV     #STACK,SP         ;RESTORE THE SP
006356  005737  001500                  TST     $PASS             ;FIRST PASS?
006362  001105                          BNE     $EOT              ;BRANCH IF NO
006364  032737  040000  177570          BIT     #SW14,@#SWR       ;IS TEST BEING LOOPED ON?
006372  001101                          BNE     $EOT              ;BRANCH IF YES
006374  104400  070326                  TYPE    ,EM724            ;TYPE MESSAGE
006400  000137  006576                  JMP     $EOT              ;GO TO NEXT TEST
006404  005010                  6$:     CLR     (R0)              ;ENSURE MAP REG HIGH CLEAR
006406  162700  000002                  SUB     #2,R0             ;GET ADDR OF MAP REG LOW
006412  012710  140000                  MOV     #140000,(R0)      ;PUT ADDR OF PAGE 6 IN MAP REG
006416  072027  000005                  ASH     #5,R0             ;ADJUST ADDR FOR PAR6
006422  052700  170000                  BIS     #170000,R0        ;SET UNIBUS ADDR BITS
006426  010037  172354                  MOV     R0,@#KIPAR6       ;PUT IN PAGE 6 PAR
006432  012737  005701  140000          MOV     #5701,@#140000    ;PUT WORD WITH PAD PARITY IN 140000
006440  012704  170000                  MOV     #170000,R4        ;PUT MAINT REG DATA IN R4
006444  012702  177750                  MOV     #MAINT,R2         ;PUT ADDRESS OF MAINT REG IN R2
006450  012737  006472  001512          MOV     #1$,$LPERR        ;SETUP ERROR LOOP
006456  012737  006534  000000          MOV     #4$,@#0           ;SETUP LOCATION ZERO
006464  012737  006554  000114          MOV     #2$,@#CACHVEC     ;SETUP CACH VECTOR
006472  012706  001500          1$:     MOV     #STACK,SP         ;INITIALIZE THE SP
006476  052737  000001  177572          BIS     #BIT0,@#MMR0      ;TURN RELOCATION ON
006504  000401                          BR      3$                ;GO TO TEST
        006506                          LOC=.
        006504                          LOC=-3&!LOC
        006510                          LOC=LOC+4
        006510                          .=LOC
006510  010412                  3$:     MOV     R4,(R2)           ;SET BITS IN MAINT REG
006512  000240                          NOP                       ;GOOD PARITY ON ODD WORD
006514  005037  140000                  CLR     @#140000          ;EXECUTE A DATIP THRU THE
                                                                  ;MAP THAT CAUSES A PE
                                ;FAILURE, NO ABORT
006520  005012                          CLR     (R2)              ;CLEAR MAINT REG
006522  000240                          NOP
006524  005037  177572                  CLR     @#MMR0            ;TURN RELOCATION OFF
006530  104174                          ERROR   174               ;NO UNIBUS PE ABORT
006532  000410                          BR      2$
                                ;TRAPPED TO WRONG VECTOR
006534  005012                  4$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
006536  000240                          NOP
006540  005037  177572                  CLR     @#MMR0            ;TURN OFF RELOCATION
006544  012737  177777  000004          MOV     #-1,@#ERRVEC      ;CLEAR ERROR REGISTER
006552  104175                          ERROR   175               ;TRAPPED TO ZERO
                                ;TEST OK
006554  005012                  2$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
006556  000240                          NOP
006560  005037  177572                  CLR     @#MMR0            ;TURN RELOCATION OFF
006564  012737  177777  177744          MOV     #-1,@#MEMERR      ;CLEAR ERROR REG
006572  005037  172516                  CLR     @#MMR3            ;ENSURE MAP TURNED OFF
006576                          $EOT:
                                ;;*****************************************************************
                                ;*TEST 5         CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FLOATING ONES
```

```
;*
;*THIS TEST IS A TEST OF BOTH THE AMX, CPU INPUTS, AND
;*THE CACHE ERROR ADDRESS REGISTER. A SET OF ADDRESSES IS
;*GENERATED AND A MAIN MEMORY ADDRESS AND CONTROL LINE
;*PARITY ERROR IS FORCED AT EACH, THEREBY LOCKING UP
;*THE ADDRESS ON THE OUTPUT OF  THE AMX IN THE ERROR
;*ADDRESS REGISTER. THE MANNER IN WHICH THIS IS DONE
;*IS AS FOLLOWS: FIRST THE ADDRESS IS GENERATED;
;*THEN, IF IT IS A VALID ADDRESS (THAT IS, IF IT IS NOT
;*BEYOND THE LIMITS OF MEMORY AS DISPLAYED IN THE
;*SYSTEM SIZE REGISTER), THESE THREE INSTRUCTIONS ARE MOVED
;*TO THAT AREA OF MEMORY:
;*       ONE:     MOV      R1,(R2)
;*       2$:      CLR      (R2)
;*       3$:      RTS      PC
;*2$ IS THE ADDRESS BEING TESTED. THE INSTRUCTION
;*AT ONE IS GIVEN CONTROL BY A 'JSR PC'. R1 IS MADE
;*TO CONTAIN #2 AND R2 CONTAINES THE ADDRESS OF
;*THE MAINTENANCE REGISTER, SO THAT AFTER THE 'MOV R1,(R2)'
;*IS EXECUTED A PARITY ERROR SHOULD OCCUR ON THE
;*MAIN MEMORY ADDRESS AND CONTROL LINES WHEN THE
;*NEXT INSTRUCTION IS FETCHED.
;*THE ADDRESSES USED ARE GENERATED FOLLOWINT THIS PATTERN
;*                200000
;*                200002
;*                200004
;*                200010
;*                200020
;*                200040
;*                200100
;*                200200
;*                200400
;*                ETC. TO:
;*                240000
;*                300000
;*                400000
;*                400002
;*                400004
;*                400010
;*                ETC. TO:
;*                500000
;*                600000
;*                1000000
;*                1000002
;*                1000004
;*                ETC.
;*THE PATTERN CONINUES UNTIL AN ADDRESS IS GENERATED THAT
;*IS TOO LARGE.
;*MEMORY MANAGEMENT IS SET UP TO FULL 22-BIT MODE, SO
;*IF THE USER WANTS TO HAVE THE EXECUTION OF THIS
;*TEST DELETED HE CAN SIMPLY BY TURNING ON THE APPORPRIATE
;*CONSOLE SWITCH WHICH HAS BEEN DESIGNATED FOR THE
;*PURPOSE OF DELETING THE EXECUTION OF TESTS WHICH
;*MAKE USER OF MEMORY MANAGEMENT.
;*
;;****************************************************************
```

```
006576  000004                          TST5:   SCOPE
006600  012737  000020  001676                  MOV     #20,$TIMES         ;;DO 20 ITERATIONS
        000005                          X=$TN-1
                                                                           ;SET THE SKAD REGISTER
006606  012737  007504  054230                  MOV     #TST6,SKAD         ;IN CASE THE TEST ABORTS.

006614  113737  001502  001626                  MOVB    $TSTNM,$TMP0
006622  012737  054076  000114                  MOV     #SPUR,@#CACHVEC             ;INITIALLY EXPECT NO ERRORS

                                                                                    ;SEE IF THIS TEST SHOULD
                                                                                    ;BE EXECUTED. THE CONDITION
                                                                                    ;TEST IS THE DESIGNATED
006630  104422                                  MMSKIP                              ;CONSOLE SWITCH.
006632  012700  172340                          MOV     #KIPAR0,R0                  ;INITIALIZE THE KERNAL
006636  012701  077406                          MOV     #77406,R1                   ;SPACE MEMORY MANAGEMENT
006642  012702  172300                          MOV     #KIPDR0,R2                  ;REGISTERS
006646  012703  000010                          MOV     #10,R3
006652  010122                          1$:     MOV     R1,(R2)+
006654  077302                                  SOB     R3,1$
006656  005020                                  CLR     (R0)+
006660  012720  000200                          MOV     #200,(R0)+
006664  012720  000400                          MOV     #400,(R0)+
006670  012720  000600                          MOV     #600,(R0)+
006674  012720  001000                          MOV     #1000,(R0)+
006700  012720  001200                          MOV     #1200,(R0)+
006704  012720  001400                          MOV     #1400,(R0)+
006710  012710  177600                          MOV     #177600,(R0)
006714  012737  000020  172516                  MOV     #20,@#MMR3                  ;TURN ON MEMORY MANAGEMENT
006722  012737  000001  177572                  MOV     #1,@#MMR0
006730  104424                                  SIZE                                ;DETERMINE FROM THE SYSTEM
                                                                                    ;SIZE REGISTER WHAT THE
                                                                                    ;HIGHEST ADDRESSABLE WORD
                                                                                    ;OF MEMORY IS.
006732  000000                          XLOADR: .WORD   0                           ;LOW ORDER 16-BITS OF THE
006734  000000                          XHIADR: .WORD   0                           ;ADDRESS AND HIGH ORDER 6-BITS
006736  042737  000002  006732                  BIC     #2,XLOADR                   ;SET THE HIGHEST WORD MINUS TWO
                                                                                    ;IN XLOADR.

006744  012737  000014  177746                  MOV     #MOM1,@#CONTRL              ;FORCE MISSES TO BOTH GROUPS.

006752  005037  007472                          CLR     XADR3                       ;INITIALIZE STORAGE
006756  005037  007474                          CLR     XADR3+2                     ;LOCATIONS USED TO GENERATE
006762  005037  007462                          CLR     XADR1                       ;THE SERIES OF TEST ADDRESSES.
006766  012737  000001  007464                  MOV     #1,XADR1+2

006774                                  X1:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
006774  023737  007464  007474                  CMP     XADR1+2,XADR3+2 ;COMPARE THE HIGH ORDER
007002  001006                                  BNE     64$                ;PARTS OF XADR1 AND ARG2.
007004  023737  007462  007472                  CMP     XADR1,XADR3        ;COMPARE THE LOW ORDER

007012  001002                                  BNE     64$                ;PARTS.



```

```
007014  000137  007436                JMP     X11             ;THEY WERE EQUAL!

007020  103402                64$:    BLO     65$
007022  000137  007032                JMP     X2              ;THE FIRST ADDRESS IS LARGER
                                                              ;THAN THE SECOND!
007026  000137  007436        65$:    JMP     X11             ;THE FIRST IS LESS THAN THE
                                                              ;SECOND.


```

```
007032                          X2:
                                ;DOUBLE PRECISION ADDITION, UNSIGNED
007032  013737  007462  007466          MOV     XADR1,XADR2
007040  013737  007464  007470          MOV     XADR1+2,XADR2+2
007046  063737  007472  007466          ADD     XADR3,XADR2
007054  005537  007470                  ADC     XADR2+2
007060  063737  007474  007470          ADD     XADR3+2,XADR2+2




007066                          X3:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007066  023737  007470  006734          CMP     XADR2+2,XLOADR+2        ;COMPARE THE HIGH ORDER
007074  001006                          BNE     64$             ;PARTS OF XADR2 AND ARG2.
007076  023737  007466  006732          CMP     XADR2,XLOADR    ;COMPARE THE LOW ORDER

007104  001002                          BNE     64$             ;PARTS.



007106  000137  007502                  JMP     XDONE           ;THEY WERE EQUAL!

007112  103402                  64$:    BLO     65$
007114  000137  007502                  JMP     XDONE           ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
007120  000137  007124          65$:    JMP     X4              ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.

007124  012737  007124  001512  X4:     MOV     #X4,$LPERR

                                ;CONVERT THE 22-BIT ADDRESS IN XADR2 TO VIRTUAL ADDRESS
                                ;WHICH WILL RELOCATE THROUGH KIPAR6; SET UP KIPAR6;
                                ;TURN ON MEMORY MANAGEMENT; PUT THE INSTRUCTIONS:
                                ;       1$:     MOV     R1,(R2)
                                ;       2$:     CLR     (R2)
                                ;       3$:     RTS     PC
                                ;AT THE LOCATION BEING TESTED, WITH 2$=TEST ADDRESS;
                                ;PUT A PATTERN,000002, IN R1 FOR THE MAINTENANCE
                                ;REGISTER TO FORCE BAD PARITY ON THE MAIN MEMORY
                                ;ADDRESS AND CONTROL LINES.  PUT THE ADDRESS OF
                                ;THE CACHE MAINTENANCE REGISTER IN R2.  PUT THE
                                ;ADDRESS, X6, IN LOCATION CACHVEC TO TAKE CARE OF THE
                                ;WHICH IS BEING FORCED. JSR TO THE ABOVE ROUTINE,
                                ;SO THAT IF THE PARITY ERROR DOES'NT OCCUR
                                ;THE 'RTS PC', AT 3$ ABOVE, WILL HANDLE IT.

007132  013703  007466                  MOV     XADR2,R3
007136  013702  007470                  MOV     XADR2+2,R2
007142  162703  000002                  SUB     #2,R3
007146  005602                          SBC     R2

007150  010300                          MOV     R3,R0
007152  042700  177701                  BIC     #177701,R0
007156  062700  140000                  ADD     #140000,R0
```

```
007162  073227  177772                          ASHC    #-6,R2
007166  010337  172354                          MOV     R3,@#KIPAR6

007172  012737  000020  172516                  MOV     #20,@#MMR3          ;TURN ON MEMORY
007200  012737  000001  177572                  MOV     #1,@#MMR0           ;MANAGEMENT.
                                                                            ;SET UP THE TEST INSTRUCTIONS.
007206  012710  010112                          MOV     #010112,(R0)        ;010112 = 'MOV R1,(R2)'
007212  012760  005012  000002                  MOV     #005012,2(R0)       ;005012 = 'CLR (R2)'
007220  012760  000207  000004                  MOV     #000207,4(R0)       ;000207 = 'RTS PC'

007226  012701  000002                          MOV     #2,R1               ;SET UP THE REGISTERS
007232  012702  177750                          MOV     #MAINT,R2

007236  012737  007256  000114                  MOV     #X6,@#CACHVEC       ;SET UP THE PARITY ERROR
007244  000240                                  NOP                         ;TRAP VECTOR AND GO.
007246  004710                                  JSR     PC,(R0)

007250                                  X5:                                 ;NO TRAP OR ABORT OCCURRED!
                                                                                    ;MAINTENANCE FUNCTION
                                                                                    ;FOR BAD PARITY ON
007250  104022                          1$:     ERROR   22                          ;THE MAIN MEMORY ADDRESS
007252  000137  007370                          JMP     X9                          ;AND CONTROL LINES FAILED

007256                                  X6:


                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007256  023737  007470  177742                  CMP     XADR2+2,LOADRS+2        ;COMPARE THE HIGH ORDER
007264  001006                                  BNE     64$                 ;PARTS OF XADR2 AND ARG2.
007266  023737  007466  177740                  CMP     XADR2,LOADRS        ;COMPARE THE LOW ORDER

007274  001002                                  BNE     64$                 ;PARTS.



007276  000137  007314                          JMP     X7                  ;THEY WERE EQUAL!

007302  103402                          64$:    BLO     65$
007304  000137  007332                          JMP     X8                  ;THE FIRST ADDRESS IS LARGER
                                                                            ;THAN THE SECOND!
007310  000137  007332                  65$:    JMP     X8                  ;THE FIRST IS LESS THAN THE
                                                                            ;SECOND.

                                                                                    ;PARITY ERROR OCCURS.
007314  005726                          X7:     TST     (SP)+                       ;RESTORE THE STACK.
007316  022626                                  CMP     (SP)+,(SP)+                 ;AND CONTINUE SINCE
007320  012737  177777  177744                  MOV     #-1,@#MEMERR                ;THE CACHE ERROR ADDRESS
007326  000137  007370                          JMP     X9                          ;REGISTER WAS SET CORRECTLY.

007332  013737  177744  001630          X8:     MOV     @#MEMERR,$TMP1              ;REPORT VALID TEST
                                                                                    ;FAILURE.
007340  013737  177740  001634                  MOV     @#LOADRS,$TMP3
007346  013737  177742  001636                  MOV     @#HIADRS,$TMP4
007354  005726                                  TST     (SP)+
007356  022626                                  CMP     (SP)+,(SP)+
007360  104023                                  ERROR   23
007362  012737  177777  177744                  MOV     #-1,@#MEMERR
```

```

 007370  005037  177572          X9:     CLR     @#MMR0                          ;TURN OFF MEMORY MANAGEMENT.
 007374  005037  172516                  CLR     @#MMR3
 007400  005737  007472                  TST     XADR3
 007404  001007                          BNE     X10                             ;GET READY TO GENERATE
 007406  005737  007474                  TST     XADR3+2                         ;THE NEXT TEST ADDRESS.
 007412  001004                          BNE     X10
 007414  012737  000002  007472          MOV     #2,XADR3
 007422  000415                          BR      X12
```

```
007424  006337  007472          X10:    ASL     XADR3
007430  006137  007474                  ROL     XADR3+2
007434  000410                          BR      X12

007436  006337  007462          X11:    ASL     XADR1
007442  006137  007464                  ROL     XADR1+2
007446  005037  007472                  CLR     XADR3
007452  005037  007474                  CLR     XADR3+2
007456  000137  006774          X12:    JMP     X1

007462  000000                  XADR1:  .WORD   0
007464  000000                          .WORD   0
007466  000000                  XADR2:  .WORD   0
007470  000000                          .WORD   0
007472  000000                  XADR3:  .WORD   0
007474  000000                          .WORD   0
007476  000000                  XADR4:  .WORD   0
007500  000000                          .WORD   0
007502  104416                  XDONE:  RSET                                    ;DONE!

                                ;;*****************************************************************
                                ;*TEST 6         CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FLOATING ZEROES
                                ;*
                                ;*THIS IS ANOTHER TEST OF THE AMX WHICH IS CARRIED
                                ;*OUT USING THE SAME METHOD AS IN THE PREVIOUS TEST
                                ;*ALL THAT IS DIFFERENT IS THE SERIES OF TEST ADDRESSES
                                ;*WHICH IS USED. IN THE PREVIOUS TEST A ONE WAS
                                ;*FLOATED THROUGH A FIELD OF ZEROES TO PRODUCE THE
                                ;*TEST ADDRESSES, HERE A ZERO WIL BE FLOATED THROUGH
                                ;*A FIELD OF ONES TO PRODUCE THE ADDRESSES
                                ;*BASE ADDRESSES WHICH ARE USE ARE:
                                ;*              177776
                                ;*              377776
                                ;*              777776
                                ;*              1777776
                                ;*              3777776
                                ;*              7777776
                                ;*              17777776
                                ;*EACH OF THESE PATTERNS IS TAKEN AND A ZERO IS FLOATED
                                ;*THROUGHT THE FIELD OF ONES TO PRODUCE A TEST ADDRESS.
                                ;*
                                ;;*****************************************************************
007504  000004                  TST6:   SCOPE
007506  012737  000020  001676          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000006                  XX=$TN-1
                                                                ;SET THE SKAD REGISTER
007514  012737  010404  054230          MOV     #TST7,SKAD      ;IN CASE THE TEST ABORTS.

007522  113737  001502  001626          MOVB    $TSTNM,$TMP0
007530  012737  054076  000114          MOV     #SPUR,@#CACHVEC                 ;INITIALLY EXPECT NO ERRORS.

007536  104422                          MMSKIP                                  ;THIS TEST MAKES USE OF
                                                                                ;MEMORY MANAGEMENT SO SEE
                                                                                ;IF THE USER HAS SET THE
                                                                                ;SWITCH DESIGNATED AS
                                                                                ;THE DON'T USE MEMORY
```

```
                                                                                     ;MANAGEMENT SWITCH.
 007540  012700  172340                     MOV     #KIPAR0,R0                       ;INITIALIZE THE KERNAL MODE
 007544  012701  077406                     MOV     #77406,R1                        ;MEMORY MANAGEMENT REGISTERS.
 007550  012702  172300                     MOV     #KIPDR0,R2
 007554  012703  000010                     MOV     #10,R3
 007560  010122                     1$:     MOV     R1,(R2)+
 007562  077302                             SOB     R3,1$
 007564  005020                             CLR     (R0)+
 007566  012720  000200                     MOV     #200,(R0)+
 007572  012720  000400                     MOV     #400,(R0)+
 007576  012720  000600                     MOV     #600,(R0)+
 007602  012720  001000                     MOV     #1000,(R0)+
 007606  012720  001200                     MOV     #1200,(R0)+
 007612  012720  001400                     MOV     #1400,(R0)+
 007616  012710  177600                     MOV     #177600,(R0)
 007622  012737  000020  172516             MOV     #20,@#MMR3                       ;TRUN ON MEMORY MANAGEMENT
 007630  012737  000001  177572             MOV     #1,@#MMR0
 007636  104424                             SIZE                            ;GET THE LARGEST MEMORY
 007640  000000                     XXLOA:  .WORD   0                                ;WORD ADDRESS INTO XXLOA
 007642  000000                     XXHIA:  .WORD   0                                ;AND XXHIA.
 007644  042737  000002  007640             BIC     #2,XXLOA                         ;GET THE ADDRESS OF THE HIGHEST WORD
                                                                                     ;WORD MINUS TWO.
 2026
```

```
007652  012737  000014  177746            MOV     #MOM1,@#CONTRL              ;FROM NOW ON FORCE MISSES
                                                                              ;TO BOTH GROUPS.

007660  012737  177776  010362  XX1:      MOV     #177776,XXADR1              ;INITIALIZE
007666  005037  010364                    CLR     XXADR1+2
007672  012704  000016                    MOV     #16,R4
007676  000410                            BR      XX3

007700  005204                  XX2:      INC     R4                          ;TURN ON THE NEXT BIT
007702  052737  000001  010362            BIS     #1,XXADR1                   ;IN THE FIELD OF ONES.
007710  006337  010362                    ASL     XXADR1
007714  006137  010364                    ROL     XXADR1+2

007720  012737  000002  010372  XX3:      MOV     #2,XXMASK                   ;INITIALIZE THE MASK
007726  005037  010374                    CLR     XXMASK+2                    ;USED TO CREATE THE ZERO
                                                                              ;IN THE FIELD OF ONES.
007732  010405                            MOV     R4,R5
007734  012737  007742  001512            MOV     #XX4,$LPERR

007742  013737  010362  010366  XX4:      MOV     XXADR1,XXADR2               ;DETERMINE THIS TEST ADDRESS.
007750  013737  010364  010370            MOV     XXADR1+2,XXADR2+2
007756  043737  010372  010366            BIC     XXMASK,XXADR2
007764  043737  010374  010370            BIC     XXMASK+2,XXADR2+2


                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007772  023737  010370  010400            CMP     XXADR2+2,XXCNST+2           ;COMPARE THE HIGH ORDER
010000  001006                            BNE     64$                 ;PARTS OF XXADR2 AND ARG2.
010002  023737  010366  010376            CMP     XXADR2,XXCNST       ;COMPARE THE LOW ORDER

010010  001002                            BNE     64$                 ;PARTS.



010012  000137  010030                    JMP     XX5                 ;THEY WERE EQUAL!

010016  103402                  64$:      BLO     65$
010020  000137  010030                    JMP     XX5                 ;THE FIRST ADDRESS IS LARGER
                                                                      ;THAN THE SECOND!
010024  000137  010320          65$:      JMP     XX10                ;THE FIRST IS LESS THAN THE
                                                                      ;SECOND.


010030                          XX5:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010030  023737  010370  007642            CMP     XXADR2+2,XXLOA+2            ;COMPARE THE HIGH ORDER
010036  001006                            BNE     64$                 ;PARTS OF XXADR2 AND ARG2.
010040  023737  010366  007640            CMP     XXADR2,XXLOA        ;COMPARE THE LOW ORDER

010046  001002                            BNE     64$                 ;PARTS.



010050  000137  010066                    JMP     XX6                 ;THEY WERE EQUAL!

```

```
010054  103402                  64$:    BLO     65$
010056  000137  010320                  JMP     XX10              ;THE FIRST ADDRESS IS LARGER
                                                                  ;THAN THE SECOND!
010062  000137  010066          65$:    JMP     XX6               ;THE FIRST IS LESS THAN THE
                                                                  ;SECOND.


010066                          XX6:

                                ;CONVERT THE 22-BIT ADDRESS IN XXADR2 TO VIRTUAL ADDRESS
                                ;WHICH WILL RELOCATE THROUGH KIPAR6; SET UP KIPAR6;
                                ;TURN ON MEMORY MANAGEMENT; PUT THE INSTRUCTIONS:
                                ;       1$:     MOV     R1,(R2)
                                ;       2$:     CLR     (R2)
                                ;       3$:     RTS     PC
                                ;AT THE LOCATION BEING TESTED, WITH 2$=TEST ADDRESS;
                                ;PUT A PATTERN,000002, IN R1 FOR THE MAINTENANCE
                                ;REGISTER TO FORCE BAD PARITY ON THE MAIN MEMORY
                                ;ADDRESS AND CONTROL LINES.  PUT THE ADDRESS OF
                                ;THE CACHE MAINTENANCE REGISTER IN R2.  PUT THE
                                ;ADDRESS, XX7, IN LOCATION CACHVEC TO TAKE CARE OF THE
                                ;WHICH IS BEING FORCED. JSR TO THE ABOVE ROUTINE,
                                ;SO THAT IF THE PARITY ERROR DOES'NT OCCUR
                                ;THE 'RTS PC', AT 3$ ABOVE, WILL HANDLE IT.

010066  013703  010366                  MOV     XXADR2,R3
010072  013702  010370                  MOV     XXADR2+2,R2
010076  162703  000002                  SUB     #2,R3
010102  005602                          SBC     R2

010104  010300                          MOV     R3,R0
010106  042700  177701                  BIC     #177701,R0
010112  062700  140000                  ADD     #140000,R0
010116  073227  177772                  ASHC    #-6,R2
010122  010337  172354                  MOV     R3,@#KIPAR6

010126  012737  000020  172516          MOV     #20,@#MMR3        ;TURN ON MEMORY
010134  012737  000001  177572          MOV     #1,@#MMR0         ;MANAGEMENT.
                                                                  ;SET UP THE TEST INSTRUCTIONS.
010142  012710  010112                  MOV     #010112,(R0)      ;010112 = 'MOV R1,(R2)'
010146  012760  005012  000002          MOV     #005012,2(R0)     ;005012 = 'CLR (R2)'
010154  012760  000207  000004          MOV     #000207,4(R0)     ;000207 = 'RTS PC'

010162  012701  000002                  MOV     #2,R1             ;SET UP THE REGISTERS
010166  012702  177750                  MOV     #MAINT,R2

010172  012737  010210  000114          MOV     #XX7,@#CACHVEC    ;SET UP THE PARITY ERROR
010200  000240                          NOP                       ;TRAP VECTOR AND GO.
010202  004710                          JSR     PC,(R0)

                                                                  ;NO TRAP OCCURRED!
010204  104024                  1$:     ERROR   24
010206  000444                          BR      XX10
                                ;COME HERE ON THE PARITY ERROR
010210                          XX7:
```

```
                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010210 023737 010370 177742                CMP     XXADR2+2,LOADRS+2      ;COMPARE THE HIGH ORDER
010216 001006                              BNE     64$              ;PARTS OF XXADR2 AND ARG2.
010220 023737 010366 177740                CMP     XXADR2,LOADRS    ;COMPARE THE LOW ORDER

010226 001002                              BNE     64$              ;PARTS.



010230 000137 010246                       JMP     XX8              ;THEY WERE EQUAL!

010234 103402                   64$:       BLO     65$
010236 000137 010262                       JMP     XX9              ;THE FIRST ADDRESS IS LARGER
                                                                    ;THAN THE SECOND!
010242 000137 010262            65$:       JMP     XX9              ;THE FIRST IS LESS THAN THE
                                                                    ;SECOND.


010246 005726                   XX8:       TST     (SP)+                    ;RESTORE THE STACK.
010250 022626                              CMP     (SP)+,(SP)+
010252 012737 177777 177744                MOV     #-1,@#MEMERR             ;RESET THE CACHE ERROR REGISTERS.
010260 000417                              BR      XX10
010262 013737 177744 001630     XX9:       MOV     @#MEMERR,$TMP1           ;REPORT A VALID TEST
                                                                    ;FAILURE.
010270 013737 177740 001634                MOV     @#LOADRS,$TMP3
010276 013737 177742 001636                MOV     @#HIADRS,$TMP4
010304 005726                              TST     (SP)+
010306 022626                              CMP     (SP)+,(SP)+
010310 104025                              ERROR   25
010312 012737 177777 177744                MOV     #-1,@#MEMERR

010320 006337 010372            XX10:      ASL     XXMASK                   ;ROTATE THE MASK.
010324 006137 010374                       ROL     XXMASK+2
010330 005305                              DEC     R5
010332 001402                              BEQ     1$
010334 000137 007742                       JMP     XX4
010340 005037 177572            1$:        CLR     @#MMR0                   ;TURN OF MEMORY MANAGEMENT.
010344 005037 172516                       CLR     @#MMR3
010350 020427 000025                       CMP     R4,#25
010354 002012                              BGE     XX11
010356 000137 007700                       JMP     XX2

010362 000000                   XXADR1:    .WORD   0                        ;USED TO GENERATE TEST PATTERNS.
010364 000000                              .WORD   0
010366 000000                   XXADR2:    .WORD   0                        ;USED TO STORE THE CURRENT
010370 000000                              .WORD   0                        ;TEST PATTERN DURING A TEST.
010372 000000                   XXMASK:    .WORD   0                        ;MASK USED TO PUT A ZERO
010374 000000                              .WORD   0                        ;IN THE FIELD OF ONES
                                                                            ;TO CREATE A TEST ADDRESS.
010376 124704                   XXCNST:    .WORD   BOTPRG                   ;THE SMALLEST ADDRESS
010400 000000                              .WORD   0                        ;IN MEMORY OVER THIS TEST.

010402 104416                   XX11:      RSET


                                ;;*****************************************************************
                                ;*TEST 7           CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ONES
```

```
                                        ;*
                                        ;*THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX.
                                        ;*THIS TEST IS IDENTICAL TO TST5 IN EVERY THING
                                        ;*IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE
                                        ;*REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM
                                        ;*THE CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL
                                        ;*GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS
                                        ;*WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE
                                        ;*AS UNIBUS REFERENCES.
                                        ;*
                                        ;;*****************************************************************
010404  000004                          TST7:   SCOPE
010406  012737  000020  001676                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000007                          RR=$TN-1
                                                                        ;SET THE SKAD REGISTER
010414  012737  011316  054230                  MOV     #TST10,SKAD     ;IN CASE THE TEST ABORTS.

010422  113737  001502  001626                  MOVB    $TSTNM,$TMP0
010430  012737  054076  000114                  MOV     #SPUR,@#CACHVEC         ;INITIALLY EXPECT NO ERRORS.
010436  012737  054050  000004                  MOV     #CPSPUR,@#ERRVEC

010444  104422                                  MMSKIP

010446  012700  172340                          MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
010452  012701  077406                          MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
010456  012702  172300                          MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
010462  012703  000010                          MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
010466  010122                          64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
010470  077302                                  SOB     R3,64$          ;ADDRESSES.
010472  005020                                  CLR     (R0)+
010474  012720  000200                          MOV     #200,(R0)+
010500  012720  000400                          MOV     #400,(R0)+
010504  012720  000600                          MOV     #600,(R0)+
010510  012720  001000                          MOV     #1000,(R0)+
010514  012720  001200                          MOV     #1200,(R0)+
010520  012720  001400                          MOV     #1400,(R0)+
010524  012710  177600                          MOV     #177600,(R0)

010530  012737  000060  172516                  MOV     #60,@#MMR3              ;TURN ON MEMORY MANAGEMENT.
010536  012737  000001  177572                  MOV     #1,@#MMR0
                                                                                ;DETERMINE THE MEMORY
010544  104424                                  SIZE                            ;SYSTEM SIZE.
010546  000000                          RRLOAD: .WORD   0                       ;LOW ORDER 16-BITS AND
010550  000000                          RRHIAD: .WORD   0                       ;HIGH ORDER 6-BITS OF THE
                                                                                ;HIGHEST MEMORY WORD ADDRESS.
010552  042737  000002  010546                  BIC     #2,RRLOAD               ;GET THE HIGHEST WORD IN MEMORY
                                                                                ;MINUS TWO.
010560  012737  000014  177746                  MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS

010566  005037  011310                          CLR     RRADR3                  ;INITIALIZE STORAGE LOCATIONS
010572  005037  011312                          CLR     RRADR3+2                ;USED TO GENERATE THE
010576  005037  011300                          CLR     RRADR1          ;SERIES OF TEST ADDRESSES.
010602  012737  000001  011302                  MOV     #1,RRADR1+2

010610                                  RR1:

```

```
                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010610  023737  011302  011312                  CMP     RRADR1+2,RRADR3+2       ;COMPARE THE HIGH ORDER
010616  001006                                  BNE     64$             ;PARTS OF RRADR1 AND ARG2.
010620  023737  011300  011310                  CMP     RRADR1,RRADR3   ;COMPARE THE LOW ORDER

010626  001002                                  BNE     64$             ;PARTS.



010630  000137  011254                          JMP     RR11            ;THEY WERE EQUAL!

010634  103402                          64$:    BLO     65$
010636  000137  010646                          JMP     RR2             ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND!
010642  000137  011254                  65$:    JMP     RR11            ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.


010646                                  RR2:
                                        ;DOUBLE PRECISION ADDITION, UNSIGNED
010646  013737  011300  011304                  MOV     RRADR1,RRADR2
010654  013737  011302  011306                  MOV     RRADR1+2,RRADR2+2
010662  063737  011310  011304                  ADD     RRADR3,RRADR2
010670  005537  011306                          ADC     RRADR2+2
010674  063737  011312  011306                  ADD     RRADR3+2,RRADR2+2




010702                                  RR3:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010702  023737  011306  010550                  CMP     RRADR2+2,RRLOAD+2       ;COMPARE THE HIGH ORDER
010710  001006                                  BNE     64$             ;PARTS OF RRADR2 AND ARG2.
010712  023737  011304  010546                  CMP     RRADR2,RRLOAD   ;COMPARE THE LOW ORDER

010720  001002                                  BNE     64$             ;PARTS.



010722  000137  011314                          JMP     RRDONE          ;THEY WERE EQUAL!

010726  103402                          64$:    BLO     65$
010730  000137  011314                          JMP     RRDONE          ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND!
010734  000137  010740                  65$:    JMP     RR4             ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.

010740  012737  010740  001512          RR4:    MOV     #RR4,$LPERR
                                        ;CONVERT THE PHYSICAL 22-BIT, ADDRESS IN RRADR2 TO A VIRTUAL ADDRESS
                                        ;WHICH WILL RELOCATE THROUGH KIPAR6 TO THE UNIBUS, THEN THROUGH
                                        ;THE MAPPING BOX TO THE UNIBUS INPUTS OF THE CACHE AMX.
010746  013737  011304  170200                  MOV     RRADR2,@#MAPL00 ;SET UP THE MAP REGISTER 0.
010754  013737  011306  170202                  MOV     RRADR2+2,@#MAPH00
010762  162737  000002  170200                  SUB     #2,@#MAPL00
010770  005637  170202                          SBC     @#MAPH00
```

```

010774 012700 140000                MOV    #140000,R0       ;A VIRTUAL ADDRESS WHICH WILL
                                                            ;RELOCATE THROUGH KIPAR6.
011000 012737 170000 172354         MOV    #170000,@#KIPAR6;RELOCATE TO UNIBUS BASE
                                                            ;ADDRESS OF 000000.
011006 012737 000060 172516         MOV    #60,@#MMR3       ;TURN ON THE MAPPING BOX AND
                                                            ;22-BIT MODE.
011014 012737 000001 177572         MOV    #1,@#MMR0        ;TURN ON MEMORY MANAGEMENT.
                                                            ;SET UP THE TEST CODE:
011022 012710 010112                MOV    #010112,(R0)     ;010112='MOV R1,(R2)'
011026 012760 005012 000002         MOV    #005012,2(R0)    ;005012='CLR (R2)'
011034 012760 000207 000004         MOV    #000207,4(R0)    ;000207='RTS PC'

011042 012701 000002                MOV    #2,R1            ;SET UP THE REGISTERS USED
011046 012702 177750                MOV    #MAINT,R2        ;IN THE TEST INSTRUCTIONS.

011052 012737 011072 000114         MOV    #RR6,@#CACHVEC   ;SET UP THE PARITY TRAP
011060 000240                       NOP                     ;VECTOR.
011062 004710                       JSR    PC,(R0) ;AND GO.


011064                       RR5:                           ;NO TRAP OR ABORT OCCURRED!
                                                                    ;MAINTENANCE FUNCTION FOR
011064 104030                1$:    ERROR  30                       ;FORCING BAD PARITY ON
011066 000137 011206                JMP    RR9                      ;THE MAIN MEMORY ADDRESS
                                                                    ;AND CONTROL LINES FAILED.
                             ;COME HERE WHEN THE FORCED ERROR OCCURS.
011072                       RR6:

                             ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
011072 023737 011306 177742         CMP    RRADR2+2,LOADRS+2        ;COMPARE THE HIGH ORDER
011100 001006                       BNE    64$              ;PARTS OF RRADR2 AND ARG2.
011102 023737 011304 177740         CMP    RRADR2,LOADRS    ;COMPARE THE LOW ORDER

011110 001002                       BNE    64$              ;PARTS.



011112 000137 011130                JMP    RR7              ;THEY WERE EQUAL!

011116 103402                64$:   BLO    65$
011120 000137 011150                JMP    RR8              ;THE FIRST ADDRESS IS LARGER
                                                            ;THAN THE SECOND!
011124 000137 011150         65$:   JMP    RR8              ;THE FIRST IS LESS THAN THE
                                                            ;SECOND.


011130 022626                RR7:   CMP    (SP)+,(SP)+
011132 005726                       TST    (SP)+                    ;RESTORE THE STACK.
011134 022626                       CMP    (SP)+,(SP)+
011136 012737 177777 177744         MOV    #-1,@#MEMERR             ;CLEAR THE CACHE ERROR REGISTER.
011144 000137 011206                JMP    RR9

011150 013737 177744 001630  RR8:   MOV    @#MEMERR,$TMP1           ;REPORT A VALID TEST FAILURE.
011156 013737 177740 001634         MOV    @#LOADRS,$TMP3
011164 013737 177742 001636         MOV    @#HIADRS,$TMP4
```

```
011172  005726                                     TST     (SP)+
011174  022626                                     CMP     (SP)+,(SP)+
011176  104031                                     ERROR   31
011200  012737  000001  177744                     MOV     #1,@#MEMERR                 ;CLEAR THE ERROR REGISTER.
011206  005037  177572          RR9:               CLR     @#MMR0                      ;TURN OFF MEMORY MANAGEMENT.
011212  005037  172516                             CLR     @#MMR3
011216  005737  011310                             TST     RRADR3                      ;GET READY TO GENERATE THE
011222  001007                                     BNE     RR10                        ;NEXT ADDRESS TO BE TESTED.
011224  005737  011310                             TST     RRADR3
011230  001004                                     BNE     RR10
011232  012737  000002  011310                     MOV     #2,RRADR3
011240  000415                                     BR      RR12

011242  006337  011310          RR10:              ASL     RRADR3
011246  006137  011312                             ROL     RRADR3+2
011252  000410                                     BR      RR12

011254  006337  011300          RR11:              ASL     RRADR1
011260  006137  011302                             ROL     RRADR1+2
011264  005037  011310                             CLR     RRADR3
011270  005037  011312                             CLR     RRADR3+2

011274  000137  010610          RR12:              JMP     RR1

011300  000000                  RRADR1: .WORD      0                                   ;3 DOUBLE WORD LOCATIONS
011302  000000                          .WORD      0                                   ;USED TO STORE 22-BIT
011304  000000                  RRADR2: .WORD      0                                   ;ADDRESSES.
011306  000000                          .WORD      0
011310  000000                  RRADR3: .WORD      0
011312  000000                          .WORD      0

011314  104416                  RRDONE: RSET                                           ;DONE!


                                ;;*****************************************************************
                                ;*TEST 10        CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ZEROES
                                ;*
                                ;*THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX.
                                ;*THIS TEST IS IDENTICAL TO TST6 IN EVERY THING
                                ;*IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE
                                ;*REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM
                                ;*THE CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL
                                ;*GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS
                                ;*WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE
                                ;*AS UNIBUS REFERENCES.
                                ;*
                                ;;*****************************************************************
011316  000004                  TST10:  SCOPE
011320  012737  000020  001676          MOV        #20,$TIMES         ;;DO 20 ITERATIONS
        000010                  SS=$TN-1
                                                                      ;SET THE SKAD REGISTER
011326  012737  012204  054230          MOV        #TST11,SKAD        ;IN CASE THE TEST ABORTS.

011334  113737  001502  001626          MOVB       $TSTNM,$TMP0
011342  012737  054076  000114          MOV        #SPUR,@#CACHVEC             ;INITIALLY EXPECT NO ERRORS
011350  104422                          MMSKIP

```

```
011352  012700  172340                    MOV     #KIPAR0,R0         ;INITIALLY PUT MEMORY
011356  012701  077406                    MOV     #77406,R1          ;MANAGEMENT IN A 'PASSIVE'
011362  012702  172300                    MOV     #KIPDR0,R2         ;STATE, THAT IS MAP ALL
011366  012703  000010                    MOV     #10,R3             ;VIRTUAL ADDRESSES ON TO
011372  010122                   64$:     MOV     R1,(R2)+           ;THEMSELVES AS PHYSICAL
011374  077302                            SOB     R3,64$             ;ADDRESSES.
011376  005020                            CLR     (R0)+
011400  012720  000200                    MOV     #200,(R0)+
011404  012720  000400                    MOV     #400,(R0)+
011410  012720  000600                    MOV     #600,(R0)+
011414  012720  001000                    MOV     #1000,(R0)+
011420  012720  001200                    MOV     #1200,(R0)+
011424  012720  001400                    MOV     #1400,(R0)+
011430  012710  177600                    MOV     #177600,(R0)

011434  104424                            SIZE                                   ;GET THE MEMORY SIZE.
011436  000000                   SSLOAD:  .WORD   0                              ;22-BIT ADDRESS OF THE
011440  000000                   SSHIAD:  .WORD   0                              ;HIGHEST WORD IN MEMORY.
011442  042737  000002  011436            BIC     #2,SSLOAD                      ;GET THE HIGHEST WORD MINUS TWO.

011450  012737  000014  177746            MOV     #MOM1,@#CONTRL

011456  012737  177776  012162   SS1:     MOV     #177776,SSADR1                 ;INITIALIZE
011464  005037  012164                    CLR     SSADR1+2
011470  012704  000016                    MOV     #16,R4
011474  000410                            BR      SS3

011476  005204                   SS2:     INC     R4                             ;TURN ON THE NEXT BIT
011500  052737  000001  012162            BIS     #1,SSADR1                      ;IN THE FIELD OF ONES
011506  006337  012162                    ASL     SSADR1
011512  006137  012164                    ROL     SSADR1+2

011516  012737  000002  012172   SS3:     MOV     #2,SSMASK                      ;INITIALIZE THE MASK USER
011524  005037  012174                    CLR     SSMASK+2                       ;TO CREATE THE ZERO IN
                                                                                 ;IN FIELD OF ONES
011530  010405                            MOV     R4,R5
011532  012737  011540  001512            MOV     #SS4,$LPERR

011540  013737  012162  012166   SS4:     MOV     SSADR1,SSADR2                  ;DETERMINE THE TEST ADDRESS.
011546  013737  012164  012170            MOV     SSADR1+2,SSADR2+2
011554  043737  012172  012166            BIC     SSMASK,SSADR2
011562  043737  012174  012170            BIC     SSMASK+2,SSADR2+2

                                 ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
011570  023737  012170  012200            CMP     SSADR2+2,SSCNST+2      ;COMPARE THE HIGH ORDER
011576  001006                            BNE     64$                  ;PARTS OF SSADR2 AND ARG2.
011600  023737  012166  012176            CMP     SSADR2,SSCNST        ;COMPARE THE LOW ORDER

011606  001002                            BNE     64$                  ;PARTS.



011610  000137  011626                    JMP     SS5                  ;THEY WERE EQUAL!

011614  103402                   64$:     BLO     65$
011616  000137  011626                    JMP     SS5                  ;THE FIRST ADDRESS IS LARGER
```

```
                                                                           ;THAN THE SECOND!
011622  000137  012120            65$:    JMP     SS10                    ;THE FIRST IS LESS THAN THE
                                                                           ;SECOND.

011626                            SS5:

                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
011626  023737  012170  011440            CMP     SSADR2+2,SSLOAD+2       ;COMPARE THE HIGH ORDER
011634  001006                            BNE     64$                     ;PARTS OF SSADR2 AND ARG2.
011636  023737  012166  011436            CMP     SSADR2,SSLOAD           ;COMPARE THE LOW ORDER

011644  001002                            BNE     64$                     ;PARTS.



011646  000137  011664                    JMP     SS6                     ;THEY WERE EQUAL!

011652  103402                    64$:    BLO     65$
011654  000137  012120                    JMP     SS10                    ;THE FIRST ADDRESS IS LARGER
                                                                           ;THAN THE SECOND!
011660  000137  011664            65$:    JMP     SS6                     ;THE FIRST IS LESS THAN THE
                                                                           ;SECOND.


011664                            SS6:
                                  ;CONVERT THE PHYSICAL 22-BIT, ADDRESS IN SSADR2 TO A VIRTUAL ADDRESS
                                  ;WHICH WILL RELOCATE THROUGH KIPAR6 TO THE UNIBUS, THEN THROUGH
                                  ;THE MAPPING BOX TO THE UNIBUS INPUTS OF THE CACHE AMX.
011664  013737  012166  170200            MOV     SSADR2,@#MAPL00 ;SET UP THE MAP REGISTER 0.
011672  013737  012170  170202            MOV     SSADR2+2,@#MAPH00
011700  162737  000002  170200            SUB     #2,@#MAPL00
011706  005637  170202                    SBC     @#MAPH00

011712  012700  140000                    MOV     #140000,R0              ;A VIRTUAL ADDRESS WHICH WILL
                                                                           ;RELOCATE THROUGH KIPAR6.
011716  012737  170000  172354            MOV     #170000,@#KIPAR6;RELOCATE TO UNIBUS BASE
                                                                           ;ADDRESS OF 000000.
011724  012737  000060  172516            MOV     #60,@#MMR3              ;TURN ON THE MAPPING BOX AND
                                                                           ;22-BIT MODE.
011732  012737  000001  177572            MOV     #1,@#MMR0               ;TURN ON MEMORY MANAGEMENT.
                                                                           ;SET UP THE TEST CODE:
011740  012710  010112                    MOV     #010112,(R0)            ;010112='MOV R1,(R2)'
011744  012760  005012  000002            MOV     #005012,2(R0)           ;005012='CLR (R2)'
011752  012760  000207  000004            MOV     #000207,4(R0)           ;000207='RTS PC'

011760  012701  000002                    MOV     #2,R1                   ;SET UP THE REGISTERS USED
011764  012702  177750                    MOV     #MAINT,R2               ;IN THE TEST INSTRUCTIONS.

011770  012737  012044  000114            MOV     #SS8,@#CACHVEC          ;SET UP THE PARITY TRAP
011776  000240                            NOP                             ;VECTOR.
012000  004710                            JSR     PC,(R0) ;AND GO.

                                                                           ;NO TRAP OCCURRED!
012002  104032                    1$:     ERROR   32
012004  000445                            BR      SS10
                                  ;TRAP TO HERE WHEN THE ERROR OCCURS.
```

```
012006                                   SS7:

                                         ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
012006  023737  012170  177742                   CMP     SSADR2+2,LOADRS+2        ;COMPARE THE HIGH ORDER
012014  001006                                   BNE     64$                      ;PARTS OF SSADR2 AND ARG2.
012016  023737  012166  177740                   CMP     SSADR2,LOADRS            ;COMPARE THE LOW ORDER

012024  001002                                   BNE     64$                      ;PARTS.



012026  000137  012044                           JMP     SS8                      ;THEY WERE EQUAL!

012032  103402                           64$:    BLO     65$
012034  000137  012062                           JMP     SS9                      ;THE FIRST ADDRESS IS LARGER
                                                                                  ;THAN THE SECOND!
012040  000137  012062                   65$:    JMP     SS9                      ;THE FIRST IS LESS THAN THE
                                                                                  ;SECOND.


012044  022626                           SS8:    CMP     (SP)+,(SP)+
012046  005726                                   TST     (SP)+                            ;RESTORE THE STACK
012050  022626                                   CMP     (SP)+,(SP)+
012052  012737  177777  177744                   MOV     #-1,@#MEMERR                     ;CLEAR THE CACHE ERROR
012060  000417                                   BR      SS10                             ;REGISTER.

012062  013737  177744  001630           SS9:    MOV     @#MEMERR,$TMP1                   ;REPORT A VALID TEST FAILURE.
012070  013737  177740  001634                   MOV     @#LOADRS,$TMP3
012076  013737  177742  001636                   MOV     @#HIADRS,$TMP4
012104  005726                                   TST     (SP)+
012106  022626                                   CMP     (SP)+,(SP)+
012110  104033                                   ERROR   33
012112  012737  177777  177744                   MOV     #-1,@#MEMERR

012120  006337  012172                   SS10:   ASL     SSMASK                           ;ROTATE MASK TO FLOAT 0
012124  006137  012174                           ROL     SSMASK+2                         ;TO THE LEFT.
012130  005305                                   DEC     R5
012132  001402                                   BEQ     1$
012134  000137  011540                           JMP     SS4
012140  005037  177572                   1$:     CLR     @#MMR0                           ;TURN OF MEMORY MANAGEMENT
012144  005037  172516                           CLR     @#MMR3                   ;AND THE MAPPING BOX.
012150  020427  000025                           CMP     R4,#25                   ;IS THE TEST DONE?
012154  002012                                   BGE     SS11                     ;YES
012156  000137  011476                           JMP     SS2                      ;NO

012162  000000                           SSADR1: .WORD   0                        ;USED TO GENERATE THE
012164  000000                                   .WORD   0                        ;TEST ADDRESSESS.
012166  000000                           SSADR2: .WORD   0
012170  000000                                   .WORD   0
012172  000000                           SSMASK: .WORD   0
012174  000000                                   .WORD   0

012176  124704                           SSCNST: .WORD   BOTPRG                   ;CONTAINS THE ADDRESS OF
012200  000000                                   .WORD   0                        ;THE LAST WORD OF THIS PROGRAM.

012202  104416                           SS11:   RSET                             ;DONE!
```

```

                                ;;**************************************************************
                                ;*TEST 11       CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS DUAL ADDRESS TEST
                                ;*
                                ;*THIS TEST PERFORMS A DUAL ADDRESS TEST ON MEMORY LOCATED
                                ;*AT ADDRESSES LESS THAN 160000 (OCT.) OR WITHIN THE FIRST
                                ;*28K.  THE PURPOSE IS TO VARIFY THE THE AMX IS WORKING
                                ;*PROPERLY FOR THE LOW ORDER ADDRESS LINES INVOLVED.
                                ;*
                                ;;**************************************************************
012204  000004                  TST11:  SCOPE
012206  012737  000004  001676          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
        000011                  PP=$TN-1
                                                                ;SET THE SKAD REGISTER
012214  012737  012442  054230          MOV     #TST12,SKAD     ;IN CASE THE TEST ABORTS.

012222  113737  001502  001626          MOVB    $TSTNM,$TMP0
012230  012737  054076  000114          MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.

012236  012737  000014  177746  PP1:    MOV     #M1M0,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS
012244  104424                          SIZE
012246  000000                  PPLOAD: .WORD   0               ;LOW ORDER 16-BITS AND
012250  000000                  PPHIAD: .WORD   0               ;HIGH ORDER 6-BITS OF THE
                                                                ;HIGHEST WORD ADDRESS IN
                                                                ;MEMORY.
012252  012737  157776  012436          MOV     #157776,PPLIM   ;ESTABLISH THE UPPER LIMIT
012260  005737  012250                  TST     PPHIAD          ;FOR THE TEST.
012264  001007                          BNE     PP2
012266  023737  012436  012246          CMP     PPLIM,PPLOAD
012274  003403                          BLE     PP2
012276  013737  012246  012436          MOV     PPLOAD,PPLIM

012304  012700  124704          PP2:    MOV     #BOTPRG,R0      ;THE LOW LIMIT FOR THIS TEST.
012310  010020                  1$:     MOV     R0,(R0)+        ;WRITE THE ADDRESS IN THE
012312  020037  012436                  CMP     R0,PPLIM        ;ADDRESS.
012316  101774                          BLOS    1$

012320  012700  124704                  MOV     #BOTPRG,R0
012324  011001                  PP3:    MOV     (R0),R1         ;GO BACK AND READ BACK THE
012326  020001                          CMP     R0,R1           ;ADDRESS, CHECK IT AND
012330  001411                          BEQ     PP4             ;WRITE BACK THE COMPLIMENT.
012332  010037  001640                  MOV     R0,$TMP5
                                                                ;REPORT ERROR.
012336  010137  001632                  MOV     R1,$TMP2
012342  010037  001634                  MOV     R0,$TMP3
012346  005037  001636                  CLR     $TMP4
012352  104034                  1$:     ERROR   34

012354  005120                  PP4:    COM     (R0)+           ;WRITE BACK COMPLIMENT.
012356  020037  012436                  CMP     R0,PPLIM
012362  101760                          BLOS    PP3

012364  012700  124704                  MOV     #BOTPRG,R0      ;GO BACK AND CHECK
012370  011001                  PP5:    MOV     (R0),R1         ;THE COMPLIMENTED PATTERNS.
012372  010002                          MOV     R0,R2
012374  005102                          COM     R2
```

```
012376  020102                          CMP     R1,R2
012400  001411                          BEQ     PP6
012402  010237  001640                  MOV     R2,$TMP5
012406  010137  001632                  MOV     R1,$TMP2
012412  010037  001634                  MOV     R0,$TMP3
012416  005037  001636                  CLR     $TMP4
012422  104034                  1$:     ERROR   34

012424  005120                  PP6:    COM     (R0)+
012426  020037  012436                  CMP     R0,PPLIM
012432  001356                          BNE     PP5
012434  000401                          BR      PP7

012436  000000                  PPLIM:  .WORD   0

012440  104416                  PP7:    RSET                            ;DONE!



                                ;;*************************************************************
                                ;*TEST 12        CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS DUAL ADDRESS TEST
                                ;*
                                ;*THIS TEST PERFORMS A DUAL ADDRESS TEST IDENTICAL TO
                                ;*TST11, EXCEPT THAT IT IS DONE THROUGH THE MAPPING
                                ;*BOX HERE THEREBY TESTING THE UNIBUS INPUTS TO THE AMX.
                                ;*
                                ;;*************************************************************
012442  000004                  TST12:  SCOPE
012444  012737  000002  001676          MOV     #2,$TIMES               ;;DO 2 ITERATIONS
        000012                  TT=$TN-1
                                                                        ;SET THE SKAD REGISTER
012452  012737  013102  054230          MOV     #TST13,SKAD             ;IN CASE THE TEST ABORTS.

012460  113737  001502  001626          MOVB    $TSTNM,$TMP0
012466  012737  054076  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
012474  104422                          MMSKIP
012476  012737  000014  177746  TT1:    MOV     #M1M0,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
012504  104424                          SIZE
012506  000000                  TTLOAD: .WORD   0                       ;DETERMINE THE HIGHEST
012510  000000                  TTHIAD: .WORD   0                       ;WORD IN MEMORY.

012512  012737  157776  013076          MOV     #157776,TTLIM           ;DETERMINE THE UPPER LIMIT
012520  005737  012510                  TST     TTHIAD                  ;FOR THE TEST.
012524  001007                          BNE     TT2
012526  023737  013076  012506          CMP     TTLIM,TTLOAD
012534  003403                          BLE     TT2
012536  013737  012506  013076          MOV     TTLOAD,TTLIM
012544                          TT2:

012544  012700  172340                  MOV     #KIPAR0,R0              ;INITIALLY PUT MEMORY
012550  012701  077406                  MOV     #77406,R1               ;MANAGEMENT IN A 'PASSIVE'
012554  012702  172300                  MOV     #KIPDR0,R2              ;STATE, THAT IS MAP ALL
012560  012703  000010                  MOV     #10,R3                  ;VIRTUAL ADDRESSES ON TO
012564  010122                  64$:    MOV     R1,(R2)+                ;THEMSELVES AS PHYSICAL
012566  077302                          SOB     R3,64$                  ;ADDRESSES.
012570  005020                          CLR     (R0)+
012572  012720  000200                  MOV     #200,(R0)+
012576  012720  000400                   MOV     #400,(R0)+
012602  012720  000600                   MOV     #600,(R0)+
012606  012720  001000                   MOV     #1000,(R0)+
012612  012720  001200                   MOV     #1200,(R0)+
012616  012720  001400                   MOV     #1400,(R0)+
012622  012710  177600                   MOV     #177600,(R0)

012626  012737  000060  172516           MOV     #60,@#MMR3         ;TURN ON MEMORY MANAGEMENT.
012634  012737  000001  177572           MOV     #1,@#MMR0
012642  012700  124704                   MOV     #BOTPRG,R0         ;INITIALIZE A POINTER.

012646                          1$:

012646  010037  170200                   MOV     R0,@#MAPL00        ;RELOCATE THE ADDRESS IN
012652  005037  170202                   CLR     @#MAPH00           ;R0 TO THE UNIBUS,
012656  012737  170000  172354           MOV     #170000,@#KIPAR6;THROUGH THE MAPPING BOX
012664  012701  140000                   MOV     #140000,R1         ;TO THE CACHE.


012670  010011                           MOV     R0,(R1)            ;WRITE THE ADDRESS IN THE
012672  062700  000002                   ADD     #2,R0              ;ADDRESS
012676  020037  013076                   CMP     R0,TTLIM
012702  101761                           BLOS    1$

012704  012700  124704                   MOV     #BOTPRG,R0

012710                          TT3:

012710  010037  170200                   MOV     R0,@#MAPL00        ;RELOCATE THE ADDRESS IN
012714  005037  170202                   CLR     @#MAPH00           ;R0 TO THE UNIBUS,
012720  012737  170000  172354           MOV     #170000,@#KIPAR6;THROUGH THE MAPPING BOX
012726  012701  140000                   MOV     #140000,R1         ;TO THE CACHE.


012732  011102                           MOV     (R1),R2            ;READ BACK THE ADDRESS
012734  020002                           CMP     R0,R2              ;AS DATA IN THE LOCATION
012736  001411                           BEQ     TT4                ;IT ADDRESSES.
012740  010037  001640                   MOV     R0,$TMP5           ;REPORT ERROR IF NOT
                                                                    ;EQUAL.
012744  010237  001632                   MOV     R2,$TMP2
012750  010037  001634                   MOV     R0,$TMP3
012754  005037  001636                   CLR     $TMP4
012760  104035                  1$:      ERROR   35
012762  005111                  TT4:     COM     (R1)               ;WRITE BACK THE
012764  062700  000002                   ADD     #2,R0              ;COMPLIMENTED DATA.
012770  020037  013076                   CMP     R0,TTLIM
012774  101745                           BLOS    TT3

012776  012700  124704                   MOV     #BOTPRG,R0

013002                          TT5:

013002  010037  170200                   MOV     R0,@#MAPL00        ;RELOCATE THE ADDRESS IN
013006  005037  170202                   CLR     @#MAPH00           ;R0 TO THE UNIBUS,
013012  012737  170000  172354           MOV     #170000,@#KIPAR6;THROUGH THE MAPPING BOX
013020  012701  140000                   MOV     #140000,R1         ;TO THE CACHE.
```

```


 013024  011102                             MOV     (R1),R2         ;GO BACK AND CHECK
 013026  010003                             MOV     R0,R3           ;THE COMPLIMENTED PATTERNS.
 013030  005103                             COM     R3
 013032  020203                             CMP     R2,R3
 013034  001411                             BEQ     TT6
 013036  010337  001640                     MOV     R3,$TMP5        ;REPORT ERROR
 013042  010237  001632                     MOV     R2,$TMP2
 013046  010037  001634                     MOV     R0,$TMP3
 013052  005037  001636                     CLR     $TMP4
 013056  104035                     1$:     ERROR   35

 013060  005111                     TT6:    COM     (R1)            ;COMPLIMENT BACK THE DATA.
 013062  062700  000002                     ADD     #2,R0
 013066  020037  013076                     CMP     R0,TTLIM
 013072  001343                             BNE     TT5
 013074  000401                             BR      TT7

 013076  000000                     TTLIM:  .WORD   0

 013100  104416                     TT7:    RSET                    ;DONE!


                                    ;;*******************************************************************
                                    ;*TEST 13         CACHE ADDRESS MEMORY COMPARATOR TEST
                                    ;*
                                    ;*THIS IS A TEST OF THE CACHE ADDRESS MEMORY ADDRESS COMPARATORS.
                                    ;*THIS IS A CIRCUIT MADE UP OF SIX 74585 CHIPS, THREE FOR EACH
                                    ;*GROUP.  EACH CHIP COMPARES FOUR BITS OF THE ADDRESS ON THE
                                    ;*ADDRESS MULTIPLEXER, AMX, OUTPUT LINES WITH THE RESPECTIVE
                                    ;*FOUR BITS FROM THE CACHE ADDRESS MEMORY.  TWELVE BITS OF
                                    ;*THE ADDRESS ARE BROKEN DOWN THUS:  BITS 10 THROUGH 13
                                    ;*FOR THE FIRST COMPARATOR; BITS 14 THROUGH 17 FOR
                                    ;*THE NEXT; AND BITS 18 THROUGH 21 FOR THE LAST.
                                    ;*THE METHOD CHOSEN FOR THIS TEST IS TO TAKE EACH
                                    ;*POSSIBLE 4-BIT INPUT CONDITION FOR A COMPARATOR FROM THE
                                    ;*ADDRESS MEMORY AND PUT EVERY POSSIBLE 4-BIT COMBINATION
                                    ;*ON THE AMX SIDE OF THE COMPARATOR.  FOR 4-BITS
                                    ;*THERE ARE 16 (DEC) CONDITIONS.  THUS FOR EVERY 4-BIT
                                    ;*ADDRESS MEMORY INPUT TO THE COMPARATOR THERE ARE
                                    ;*16 AMX INPUT COMBINATIONS ONE OF WHICH WILL CAUSE
                                    ;*A MATCH AND MAKE THE REFERENCE A HIT.  THE OTHER
                                    ;*15 SHOULD OF COURSE BE MISSES.
                                    ;*
                                    ;;*******************************************************************
 013102  000004                     TST13:  SCOPE
 013104  012737  000040  001676             MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                                                    ;SET THE SKAD REGISTER
 013112  012737  014254  054230             MOV     #TST14,SKAD     ;IN CASE THE TEST ABORTS.

 013120  113737  001502  001626             MOVB    $TSTNM,$TMP0
 013126  012737  054076  000114             MOV     #SPUR,@#CACHVEC

 013134  104422                             MMSKIP                  ;SEE IF THE SWITCH REGISTER
                                                                    ;REFLECTS THE USERS DESIRE
                                                                    ;TO ELIMINATE EXECUTION OF ANY TESTS
                                                                    ;USING MEMORY MANAGEMENT.  IF
                                                                    ;SO GO TO THE NEXT TEST.

013136  012700  172340                  MOV     #KIPAR0,R0          ;INITIALLY PUT MEMORY
013142  012701  077406                  MOV     #77406,R1           ;MANAGEMENT IN A 'PASSIVE'
013146  012702  172300                  MOV     #KIPDR0,R2          ;STATE, THAT IS MAP ALL
013152  012703  000010                  MOV     #10,R3              ;VIRTUAL ADDRESSES ON TO
013156  010122                  64$:    MOV     R1,(R2)+            ;THEMSELVES AS PHYSICAL
013160  077302                          SOB     R3,64$              ;ADDRESSES.
013162  005020                          CLR     (R0)+
013164  012720  000200                  MOV     #200,(R0)+
013170  012720  000400                  MOV     #400,(R0)+
013174  012720  000600                  MOV     #600,(R0)+
013200  012720  001000                  MOV     #1000,(R0)+
013204  012720  001200                  MOV     #1200,(R0)+
013210  012720  001400                  MOV     #1400,(R0)+
013214  012710  177600                  MOV     #177600,(R0)


013220  104424                          SIZE
013222  000000                  ZADLO:  .WORD   0                   ;THE HIGHEST ADDRESSABLE
013224  000000                  ZADHI:  .WORD   0                   ;MEMORY WORD AVAILABLE.

013226  005037  014020                  CLR     ZFLG1               ;ZFLG1 INDICATES WHICH GROUP
                                                                    ;IS BEING TESTED.
                                                                    ;ZFLG1 = 0, TESTING GROUP 0.
                                                                    ;ZFLG1 = 1, TESTING GROUP 1.
                                                                    ;TEST GROUP 0 FIRST.

013232  012737  000030  014026          MOV     #S0M1,ZGS           ;ZGS AND ZGM CONTAIN
013240  012737  000044  014024          MOV     #S1M0,ZGM           ;PATTERNS TO BE USED IN
                                                                    ;THE CACHE CONTROL REGISTER.
013246  005037  014022                  CLR     ZFLG2               ;ZFLG2 INDICATES WHICH
                                                                    ;4-BIT ADDRESS FIELD, OR
                                                                    ;WHICH COMPARATOR, IS
                                                                    ;BEING TESTED.
                                                                    ;ZFLG2 = 0, BITS 10 THROUGH 13
                                                                    ;ZFLG2 = 1, BITS 14 THROUGH 17
                                                                    ;ZFLG2 = 2, BITS 18 THROUGH 21
                                                                    ;ZFLG2 = 3, DONE!

013252  005737  014022          Z1:     TST     ZFLG2               ;SEE WHICH COMPARATOR
013256  001010                          BNE     Z2                  ;IS BEING TESTED ON THIS
                                                                    ;PASS AND PUT THE SIXTEEN
                                                                    ;POSSIBLE ADDRESSES NEEDED
                                                                    ;FOR THE TEST IN ZTABLE.
013260  012737  002000  014046          MOV     #2000,ZTABLE+4      ;BITS 10-13
013266  005037  014050                  CLR     ZTABLE+6
013272  004737  014144                  JSR     PC,ZCMTBL           ;CALL ZCMTBL TO FINISH THE TABLE.
013276  000432                          BR      Z5

013300  022737  000001  014022  Z2:     CMP     #1,ZFLG2
013306  001010                          BNE     Z3

013310  012737  040000  014046          MOV     #40000,ZTABLE+4     ;BITS 14-17
```

```
013316  005037  014050                  CLR     ZTABLE+6
013322  004737  014144                  JSR     PC,ZCMTBL       ;GET ZCMTBL TO FINISH SETTING
013326  000416                          BR      Z5              ;UP THE TABLE.

013330  022737  000002  014022  Z3:     CMP     #2,ZFLG2
013336  001010                          BNE     Z4

013340  012737  000004  014050          MOV     #4,ZTABLE+6     ;BITS 18-21
013346  005037  014046                  CLR     ZTABLE+4
013352  004737  014144                  JSR     PC,ZCMTBL
013356  000402                          BR      Z5

013360  000137  013752          Z4:     JMP     Z14             ;DONE WITH THIS GROUP.

013364  012701  014032          Z5:     MOV     #ZTHR,R1
013370  013737  014024  177746          MOV     ZGM,@#CONTRL
013376  005711                          TST     (R1)            ;MAKE ZTHR A HIT IN BOTH GROUPS.
013400  013737  014026  177746          MOV     ZGS,@#CONTRL
013406  005711                          TST     (R1)

                                                                ;FROM NOW ON SELECT THE GROUP BEING TESTED
                                                                ;WHILE MISSING THE OTHER GROUP.
013410  012737  000020  172516          MOV     #20,@#MMR3      ;TURN ON MEMORY MANAGEMENT.
013416  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE!

013424  012701  014042                  MOV     #ZTABLE,R1      ;INITIALIZE R1 AS A POINTER
                                                                ;TO THE ADDRESS WHICH WILL
                                                                ;BE MADE A HIT.

013430                          Z7:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES


013430  023761  013224  000002          CMP     ZADLO+2,2(R1)   ;COMPARE THE HIGH ORDER
013436  001005                          BNE     64$             ;PARTS OF ZADLO AND (R1).
013440  023711  013222                  CMP     ZADLO,(R1)      ;THEN IF NECESSARY
013444  001002                          BNE     64$             ;COMPARE THE LOW ORDER PARTS.

013446  000137  013464                  JMP     1$              ;THEY WERE EQUAL!

013452  103402                  64$:    BLO     65$
013454  000137  013464                  JMP     1$              ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
013460  000137  013752          65$:    JMP     Z14             ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.


013464  012702  014042          1$:     MOV     #ZTABLE,R2      ;INITIALIZE A POINTER TO
                                                                ;THE ADDRESSES WHICH WILL
                                                                ;BE FED THROUGH THE COMPARATOR
                                                                ;AGAINST THE ADDRESS POINTED
                                                                ;TO BY THE OTHER POINTER, R1

013470  020102                  Z8:     CMP     R1,R2           ;DON'T TEST THE ADDRESS
013472  001511                          BEQ     Z12             ;AGAINST ITSELF HERE.
```

```

013474                            Z9:

                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES


013474  023762  013224  000002           CMP     ZADLO+2,2(R2)    ;COMPARE THE HIGH ORDER
013502  001005                           BNE     64$              ;PARTS OF ZADLO AND (R2).
013504  023712  013222                   CMP     ZADLO,(R2)       ;THEN IF NECESSARY
013510  001002                           BNE     64$              ;COMPARE THE LOW ORDER PARTS.

013512  000137  013530                   JMP     Z10              ;THEY WERE EQUAL!

013516  103402                    64$:   BLO     65$
013520  000137  013530                   JMP     Z10              ;THE FIRST ADDRESS IS LARGER
                                                                  ;THAN THE SECOND!
013524  000137  013730            65$:   JMP     Z13              ;THE FIRST IS LESS THAN THE
                                                                  ;SECOND.


013530                            Z10:

013530  011103                           MOV     (R1),R3          ;GET THE PHYSICAL ADDRESS POINTED
013532  042703  177700                   BIC     #177700,R3       ;TO BY R1 AND ESTABLISH
013536  011105                           MOV     (R1),R5 ;A VIRTUAL ADDRESS WHICH
013540  016104  000002                   MOV     2(R1),R4         ;WILL RELOCATE THROUGH
013544  073427  177772                   ASHC    #-6,R4           ;KIPAR6. SETUP KIPAR6 AND
013550  010537  172354                   MOV     R5,@#KIPAR6      ;LEAVE THE VIRTUAL ADDRESS
013554  062703  140000                   ADD     #140000,R3       ;IN R3.


013560  005713                           TST     (R3)
013562  005713                           TST     (R3)             ;SEE IF YOU CAN GET A HIT.
013564  032737  000010  177752           BIT     #10,@#HITMIS
013572  001011                           BNE     Z11
013574  013737  014020  001630           MOV     ZFLG1,$TMP1      ;NO! REPORT THE FAILURE
013602  011137  001632                   MOV     (R1),$TMP2
013606  016137  000002  001634           MOV     2(R1),$TMP3
013614  104026                    1$:    ERROR   26

013616                            Z11:

013616  011203                           MOV     (R2),R3          ;GET THE PHYSICAL ADDRESS POINTED
013620  042703  177700                   BIC     #177700,R3       ;TO BY R2 AND ESTABLISH
013624  011205                           MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
013626  016204  000002                   MOV     2(R2),R4         ;WILL RELOCATE THROUGH
013632  073427  177772                   ASHC    #-6,R4           ;KIPAR6. SETUP KIPAR6 AND
013636  010537  172354                   MOV     R5,@#KIPAR6      ;LEAVE THE VIRTUAL ADDRESS
013642  062703  140000                   ADD     #140000,R3       ;IN R3.


013646  000240                           NOP                      ;FOR SCOPING WITH AN OSCILLOSCOPE.
013650  005713                           TST     (R3)             ;MAKE SURE THERE IS NO
013652  032737  000010  177752           BIT     #10,@#HITMIS             ;MATCH.  A MISS?
013660  001416                           BEQ     Z12
013662  013737  014020  001630           MOV     ZFLG1,$TMP1      ;GOT A HIT! SO REPORT
```

```
013670  011137  001632                           MOV     (R1),$TMP2          ;FAILURE
013674  016137  000002  001634                   MOV     2(R1),$TMP3
013702  011237  001636                           MOV     (R2),$TMP4
013706  016237  000002  001640                   MOV     2(R2),$TMP5
013714  104027                           1$:     ERROR   27

013716  062702  000004                   Z12:    ADD     #4,R2               ;MOVE POINTER TO NEXT AMX
                                                                             ;SIDE COMPARATOR INPUT ADDRESS.
013722  020227  014142                           CMP     R2,#ZTABOT          ;DONE?
013726  001260                                   BNE     Z8                  ;BRANCH IF NOT DONE.

013730  062701  000004                   Z13:    ADD     #4,R1               ;GO TO THE NEXT ADDRESS
013734  020127  014142                           CMP     R1,#ZTABOT          ;IN THE TABLE; OR IS THE
013740  001233                                   BNE     Z7                  ;TEST USING THIS ADDRESS TABLE DONE?
                                                                             ;IF NOT GO TO Z7.
013742  005237  014022                           INC     ZFLG2               ;IF DONE WITH THESE ADDRESSES
013746  000137  013252                           JMP     Z1                  ;GO BACK TO COMPUTE THE
                                                                             ;NEXT ADDRESS TABLE, THAT IS
                                                                             ;CHECK THE NEXT 4-BIT
                                                                             ;COMPARATOR
013752  005037  177572                   Z14:    CLR     @#MMR0              ;TURN OFF MEMORY MANAGEMENT.
013756  005037  172516                           CLR     @#MMR3
013762  005737  014020                           TST     ZFLG1               ;SEE IF BOTH GROUPS HAVE
013766  001131                                   BNE     Z15                 ;BEEN TESTED. BRANCH IF YES
013770  005237  014020                           INC     ZFLG1               ;OTHERWISE CHANGE THE
013774  012737  000044  014026                   MOV     #S1M0,ZGS           ;PATTERNS USED IN THE CACHE
014002  012737  000030  014024                   MOV     #S0M1,ZGM           ;CONTROL REGISTER AND GO
014010  005037  014022                           CLR     ZFLG2               ;BACK TO TEST GROUP 1.
014014  000137  013252                           JMP     Z1

014020  000000                           ZFLG1:  .WORD   0                   ;FLAG WHICH DESIGNATES WHICH
                                                                             ;GROUP IS BEING TESTED, 0 OR 1.
014022  000000                           ZFLG2:  .WORD   0                   ;FLAG WHICH DESIGNATES WHICH
                                                                             ;COMPARATOR IS BEING TESTED:
                                                                             ;0 - BITS 10 THROUGH 13
                                                                             ;1 - BITS 14 THROUGH 17
                                                                             ;2 - BITS 18 THROUGH 21.

014024  000000                           ZGM:    .WORD   0                   ;PATTERNS USED IN THE HIT
014026  000000                           ZGS:    .WORD   0                   ;AND MISS REGISTER.
014030  000000                                   .WORD   0
014032  000000                           ZTHR:   .WORD   0
014034  000000                                   .WORD   0

014036  000000                           ZTMP1:  .WORD   0                   ;TEMPORARY STORAGE LOCATIONS
014040  000000                           ZTMP2:  .WORD   0                   ;USED BY THE ROUTINE, ZCMTBL,
                                                                             ;TO GENERATE THE TEST ADDRESS
                                                                             ;TABLE, ZTABLE.

014042  000040                           ZTABLE: .BLKW   40                  ;THE TEST ADDRESS TABLE.
014142  000000                           ZTABOT: .WORD   0                   ;PRECISION, 22-BIT, ADDRESSES.

                                         ;THIS ROUTINE IS CALLED TO GENERATE THE TEST ADDRESS
                                         ;TABLE, BY A 'JSR PC,ZCMTBL'.  IT CLEARS THE FIRST
                                         ;ENTRY; IT ASSUMES THE THE BASE ADDRESS HAS BEEN
                                         ;PLACED IN THE SECOND ENTRY BEFORE CONTROL IS PASSED
```

```
                                      ;TO IT; THEN, STARTING WITH THE THIRD ENTRY, IT COMPUTES
                                      ;EACH ENTRY BY ADDING THE BASE ADDRESS TO THE PRECEEDING
                                      ;ENTRY.
014144  012701  014042                ZCMTBL: MOV     #ZTABLE,R1          ;ESTABLISH A POINTER TO
                                                                          ;THE TABLE.
014150  005021                                CLR     (R1)+               ;CLR THE FIRST ENTRY.
014152  005021                                CLR     (R1)+
014154  012700  000016                        MOV     #16,R0
014160  012137  014036                1$:     MOV     (R1)+,ZTMP1         ;SAVE THE CURRENT ENTRY
014164  012137  014040                        MOV     (R1)+,ZTMP2
                                                                          ;ADD THE OFFSET TO THE
                                      ;DOUBLE PRECISION ADDITION, UNSIGNED



014170  013711  014036                        MOV     ZTMP1,(R1)
014174  013761  014040  000002                MOV     ZTMP1+2,2(R1)
014202  063711  014046                        ADD     ZTABLE+4,(R1)
014206  005561  000002                        ADC     2(R1)
014212  063761  014050  000002                ADD     ZTABLE+4+2,2(R1)
014220  077021                                SOB     R0,1$               ;LOOP UNTIL ZTABLE IS FILLED.


014222  012702  000020                        MOV     #20,R2
014226  012701  014042                        MOV     #ZTABLE,R1
014232  012700  014032                        MOV     #ZTHR,R0
014236  042700  176000                        BIC     #176000,R0
014242  060021                        2$:     ADD     R0,(R1)+
014244  005721                                TST     (R1)+
014246  077203                                SOB     R2,2$

014250  000207                                RTS     PC                  ;THE RETURN

014252  104416                        Z15:    RSET                        ;DONE!



                                      ;;*****************************************************************
                                      ;*TEST 14         CACHE ADDRESS MEMORY COUNT PATTERN TEST
                                      ;*
                                      ;*THIS IS A TEST OF THE ADDRESS MEMORY IN THE CACHE.
                                      ;*EVERY BIT IN THE MEMORY IS TURNED ON AND OFF WITHIN
                                      ;*THE LIMITATIONS OF MEMORY SIZE.  THE MANNER IN WHICH
                                      ;*THIS IS DONE IS TO ATTEMPT TO MAKE EVERY ADDRESS
                                      ;*IN AVAILABLE MEMORY A HIT IN EACH GROUP.
                                      ;*
                                      ;;*****************************************************************
014254  000004                        TST14:  SCOPE
014256  012737  000002  001676                MOV     #2,$TIMES           ;;DO 2 ITERATIONS
        000014                        BB=$TN-1
014264                                BB0:
                                                                          ;SET THE SKAD REGISTER
014264  012737  015304  054230                MOV     #TST15,SKAD         ;IN CASE THE TEST ABORTS.

014272  113737  001502  001626                MOVB    $TSTNM,$TMP0

014300  104422                                MMSKIP
```

```
3091
3092  014302  104424                             SIZE
3093  014304  000000                    BBLOAD:  .WORD    0
3094  014306  000000                    BBHIAD:  .WORD    0
3095
3096  014310  005037  015006                     CLR      BBFLG1              ;TEST GROUP 0 FIRST.
3097  014314  012737  000034  015016             MOV      #S0M0M1,BBGS
3098  014322  012737  000054  015020             MOV      #S1M0M1,BBGM
3099
3100  014330  012737  054076  000114  BB1:       MOV      #SPUR,@#CACHVEC   ;EXPECT NO ERRORS, FOR NOW.
3101  014336  012700  014264                     MOV      #BB0,R0             ;MAKE THIS CODE HITS IN
3102  014342  012701  001000                     MOV      #1000,R1            ;THE GROUP NOT BEING TESTED.
3103  014346  013737  015016  177746  BB2:       MOV      BBGS,@#CONTRL
3104  014354  005760  002000                     TST      2000(R0)
3105  014360  013737  015020  177746             MOV      BBGM,@#CONTRL
3106  014366  005720                             TST      (R0)+
3107  014370  077112                             SOB      R1,BB2
3108
3109  014372  013700  015016                     MOV      BBGS,R0             ;FROM NOW ON FORCE
3110  014376  042700  177717                     BIC      #177717,R0          ;SELECT THE GROUP BEING
3111  014402  010037  177746                     MOV      R0,@#CONTRL         ;TESTED.
3112
3113  014406  012700  014772          BB3:       MOV      #BBADR1,R0          ;INITIALIZE.
3114  014412  012720  124704                     MOV      #BOTPRG,(R0)+       ;CONTAINS THE TEST ADDRESS.
3115  014416  005020                             CLR      (R0)+
3116  014420  005020                             CLR      (R0)+               ;CONTAINS THE LOGICAL 'OR'
3117  014422  005020                             CLR      (R0)+               ;OF FAILING ADDRESSES.
3118  014424  012720  177777                     MOV      #-1,(R0)+           ;CONTAINS THE LOGICAL 'AND'
3119  014430  012720  177777                     MOV      #-1,(R0)+           ;OF BAD ADDRESSES
3120
3121
3122  014434  012700  172340                     MOV      #KIPARC,R0          ;INITIALLY PUT MEMORY
3123  014440  012701  077406                     MOV      #77406,R1           ;MANAGEMENT IN A 'PASSIVE'
3124  014444  012702  172300                     MOV      #KIPDR0,R2          ;STATE, THAT IS MAP ALL
3125  014450  012703  000010                     MOV      #10,R3              ;VIRTUAL ADDRESSES ON TO
3126  014454  010122                  64$:       MOV      R1,(R2)+            ;THEMSELVES AS PHYSICAL
3127  014456  077302                             SOB      R3,64$              ;ADDRESSES.
3128  014460  005020                             CLR      (R0)+
3129  014462  012720  000200                     MOV      #200,(R0)+
3130  014466  012720  000400                     MOV      #400,(R0)+
3131  014472  012720  000600                     MOV      #600,(R0)+
3132  014476  012720  001000                     MOV      #1000,(R0)+
3133  014502  012720  001200                     MOV      #1200,(R0)+
3134  014506  012720  001400                     MOV      #1400,(R0)+
3135  014512  012710  177600                     MOV      #177600,(R0)
3136
3137  014516  012737  000020  172516             MOV      #20,@#MMR3          ;TURN ON MEMORY MANAGEMENT.
3138  014524  012737  000001  177572             MOV      #1,@#MMR0
3139
3140  014532  005037  015010                     CLR      BBFLG2              ;INITIALIZE THE ERROR
3141  014536  005037  015012                     CLR      BBCNT1              ;FLAG AND COUNT.
3142  014542  005037  015014                     CLR      BBCNT1+2
3143
3144  014546  012737  015022  000114             MOV      #BBERR1,@#CACHVEC        ;PREPARE FOR ERRORS.
3145
3146  014554                          BB4:
```

```

                                   ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
014554  023737  014306  014774             CMP     BBLOAD+2,BBADR1+2       ;COMPARE THE HIGH ORDER
014562  001006                             BNE     64$                 ;PARTS OF BBLOAD AND ARG2.
014564  023737  014304  014772             CMP     BBLOAD,BBADR1       ;COMPARE THE LOW ORDER

014572  001002                             BNE     64$                 ;PARTS.



014574  000137  014612                     JMP     BB5                 ;THEY WERE EQUAL!

014600  103402                     64$:    BLO     65$
014602  000137  014710                     JMP     BB7                 ;THE FIRST ADDRESS IS LARGER
                                                                       ;THAN THE SECOND!
014606  000137  014612             65$:    JMP     BB5                 ;THE FIRST IS LESS THAN THE
                                                                       ;SECOND.


014612  012700  014772             BB5:    MOV     #BBADR1,R0          ;SET UP MEMORY MANAGEMENT.

014616  011003                             MOV     (R0),R3             ;GET THE PHYSICAL ADDRESS POINTED
014620  042703  177700                     BIC     #177700,R3          ;TO BY R0 AND ESTABLISH
014624  011005                             MOV     (R0),R5 ;A VIRTUAL ADDRESS WHICH
014626  016004  000002                     MOV     2(R0),R4            ;WILL RELOCATE THROUGH
014632  073427  177772                     ASHC    #-6,R4              ;KIPAR6. SETUP KIPAR6 AND
014636  010537  172354                     MOV     R5,@#KIPAR6         ;LEAVE THE VIRTUAL ADDRESS
014642  062703  140000                     ADD     #140000,R3          ;IN R3.


014646  000240                             NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
014650  005713                             TST     (R3)                ;TRY TO GET A HIT.
014652  005713                             TST     (R3)

014654  032737  000010  177752             BIT     #10,@#HITMIS        ;WAS IT A HIT?
014662  001004                             BNE     BB6                 ;BRANCH IF YES, OTHERWISE
                                                                       ;REPORT ERROR.
014664  013737  015006  001632             MOV     BBFLG1,$TMP2
014672  104036                     1$:     ERROR   36

014674  062737  000004  014772     BB6:    ADD     #4,BBADR1           ;MOVE TO NEXT WORD PAIR.
014702  005537  014774                     ADC     BBADR1+2
014706  000722                             BR      BB4

014710  005737  015010             BB7:    TST     BBFLG2              ;DID AN ERROR OCCUR IN
014714  001410                             BEQ     BB8                 ;THAT GROUP, IF YES PRINT
014716  112737  000037  001516             MOVB    #37,$ITEMB          ;AN ERROR SUMMARY
014724  013737  015006  001630             MOV     BBFLG1,$TMP1
014732  004737  054744                     JSR     PC,ERTYPE

014736  005737  015006             BB8:    TST     BBFLG1              ;HAVE BOTH GROUPS BEEN TESTED?
014742  001157                             BNE     BBDONE
014744  012737  000001  015006             MOV     #1,BBFLG1           ;IF NOT, GO BACK AND
014752  012737  000054  015016             MOV     #S1M0M1,BBGS        ;TEST GROUP 1
014760  012737  000034  015020             MOV     #S0M0M1,BBGM
014766  000137  014330                     JMP     BB1
```

```

014772  000000                  BBADR1: .WORD   0               ;THE TEST ADDRESS.
014774  000000                          .WORD   0
014776  000000                  BBADR2: .WORD   0               ;LOGICAL 'OR' OF BAD ADDRESSES.
015000  000000                          .WORD   0
015002  000000                  BBADR3: .WORD   0               ;LOGICAL 'AND' OF BAD ADDRESSES.
015004  000000                          .WORD   0

015006  000000                  BBFLG1: .WORD   0               ;FLAG: 1, IF TESTING GROUP 1,
                                                                ;OR 0, IF TESTING GROUP 0.
015010  000000                  BBFLG2: .WORD   0               ;ERROR FLAG: 0, IF NO ERRORS
                                                                ;OCCURRED IN THE TESTED
                                                                ;GROUP.
015012  000000                  BBCNT1: .WORD   0               ;ERROR COUNT.
015014  000000                          .WORD   0

015016  000000                  BBGS:   .WORD   0               ;PATTERNS FOR THE CACHE
015020  000000                  BBGM:   .WORD   0               ;CONTROL REGISTER

015022                          BBERR1:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
015022  023737  177742  014774          CMP     LOADRS+2,BBADR1+2       ;COMPARE THE HIGH ORDER
015030  001006                          BNE     64$             ;PARTS OF LOADRS AND ARG2.
015032  023737  177740  014772          CMP     LOADRS,BBADR1   ;COMPARE THE LOW ORDER

015040  001002                          BNE     64$             ;PARTS.



015042  000137  015060                  JMP     BBERR2          ;THEY WERE EQUAL!

015046  103402                  64$:    BLO     65$
015050  000137  054076                  JMP     SPUR            ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
015054  000137  054076          65$:    JMP     SPUR            ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.


015060  032737  000060  177744  BBERR2: BIT     #60,@#MEMERR    ;MAKE SURE A CACHE ADDRESS
015066  001002                          BNE     BBERR3          ;MEMORY PARITY ERROR OCCURRED.
015070  000137  054076                  JMP     SPUR

015074                          BBERR3:                         ;REPORT ERROR.
015074  013737  015006  001634          MOV     BBFLG1,$TMP3
015102  012637  001632                  MOV     (SP)+,$TMP2
015106  005726                          TST     (SP)+
015110  013737  177744  001636          MOV     @#MEMERR,$TMP4
015116  013737  177740  001644          MOV     @#LOADRS,$TMP7
015124  013737  177742  001646          MOV     @#HIADRS,$TMP10
015132  013737  014772  001640          MOV     BBADR1,$TMP5
015140  013737  014774  001642          MOV     BBADR1+2,$TMP6
015146  104040                  1$:     ERROR   40

015150  053737  014772  014776          BIS     BBADR1,BBADR2   ;COMPUTE LOGICAL 'OR' OF
015156  053737  014774  015000          BIS     BBADR1+2,BBADR2+2       ;BAD ADDRESSES.
```

```
015164  005137  015002                  COM     BBADR3          ;COMPUT THE LOGICAL 'AND'
015170  043737  014772  015002          BIC     BBADR1,BBADR3   ;OF THE BAD ADDRESSES.
015176  005137  015002                  COM     BBADR3
015202  005137  015004                  COM     BBADR3+2
015206  043737  014774  015004          BIC     BBADR1+2,BBADR3+2
015214  005137  015004                  COM     BBADR3+2

015220  012737  177777  015010          MOV     #-1,BBFLG2      ;SET THE ERROR FLAG.
015226  005237  015012                  INC     BBCNT1          ;INCREMENT THE ERROR
015232  005537  015014                  ADC     BBCNT1+2        ;COUNT.

015236  012737  015260  000114          MOV     #BBERR4,@#CACHVEC       ;TRY TO GET THE BAD
                                                                ;ADDRESS OUT OF THE ADDRESS
                                                                ;MEMORY.
015244  013705  177740                  MOV     @#LOADRS,R5
015250  042705  176001                  BIC     #176001,R5
015254  005715                          TST     (R5)
015256  000401                          BR      BBERR5
015260  022626                  BBERR4: CMP     (SP)+,(SP)+
015262  012737  177777  177744  BBERR5: MOV     #-1,@#MEMERR
015270  012737  015022  000114          MOV     #BBERR1,@#CACHVEC
015276  000137  014674                  JMP     BB6

015302  104416                  BBDONE: RSET                    ;DONE!

                                ;;****************************************************************
                                ;*TEST 15       CACHE ADDRESS MEMORY PARITY LOGIC TEST
                                ;*
                                ;*THIS IS A TEST OF THE PARITY CHECKERS AND PARITY GENERATOR
                                ;*OF THE CACHE ADDRESS MEMORY. EVERY POSSIBLE ADDRESS TAG,
                                ;*BITS 21 THROUGH 10, WHICH CAN BE STORED IN THE CACHE
                                ;*ADDRESS MEMORY IS GENERATED, MADE A HIT AND THE
                                ;*MAINTENANCE REGISTER IS THEN USED TO FORCE A CACHE ADDRESS
                                ;*MEMORY PARITY ERROR AT EACH OF THE ADDRESSES
                                ;*GENERATED. NOTE THAT BITS 9 THROUGH 0 OF THE ADDRESSES
                                ;*IS NOT OF CONCERN, SO THESE BITS WILL BE THE SAME
                                ;*FOR EACH ADDRESS; THIS IS BECAUSE ONLY BITS 21 THROUGH
                                ;*10 ARE STORED IN THE ADDRESS MEMORY THEREFORE ONLY
                                ;*THESE BITS ARE PARITY CHECKED IN THE CACHE ADDRESS
                                ;*MEMORY PARITY CHECKERS. ALSO NOTE THAT THE RANGE
                                ;*OF THE ADDRESSES MUST BE LIMITED TO BETWEEN THE
                                ;*BOUNDS IMPOSED BY THE HIGHEST AVAILABLE MEMORY WORD
                                ;*AND THE LAST WORD OF MEMORY USED BY THIS PROGRAM.
                                ;*THE MANNER IN WHICH THE ERROR WILL BE FORCED
                                ;*WILL BE TO PUT THE INSTRUCTIONS:
                                ;*      1$:     MOV     R4,(R2)
                                ;*      TSTADS: CLR     (R2)
                                ;*              RTS     PC
                                ;*AT THE PARTICULAR ADDRESS BEING TESTED, WHERE
                                ;*'TSTADS' IS THE ADDRESS BEING TESTED. R4 CONTAINS
                                ;*A PATTERN TO BE LOADED IN THE MAINTENANCE REGISTER
                                ;*WHICH WILL FORCE AN ERROR IN THE CACHE ADDRESS
                                ;*MEMORY; R2 CONTAINS THE ADDRESS OF THE MAINTENANCE
                                ;*REGISTER. NOTE FOR EACH ADDRESS R4 WILL FIRST
                                ;*BE SUCH AS TO CAUSE AN ERROR IN THE LOW
                                ;*BYTE ADDRESS PARITY CHECKER THEN AT THE SAME
```

```
                               ;*ADDRESS AN ERROR WILL BE FORCED ON THE HIGH BYTE!
                               ;*THE SEQUENCE OF TEST ADDRESSES WILL BE GENERATED
                               ;*TWICE ONCE MAKING THEM HITS IN GROUP 0 THEN
                               ;*MAKING THEM HITS IN GROUP 1.
                               ;*
                               ;;************************************************************
015304  000004                 TST15:  SCOPE
015306  012737  000002  001676         MOV     #2,$TIMES      ;;DO 2 ITERATIONS
        000015                 AA=$TN-1
                                                              ;SET THE SKAD REGISTER
015314  012737  016374  054230         MOV     #TST16,SKAD    ;IN CASE THE TEST ABORTS.

015322  113737  001502  001626         MOVB    $TSTNM,$TMP0
015330  012737  054076  000114         MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
015336  104422                         MMSKIP

015340  012700  172340                 MOV     #KIPAR0,R0     ;INITIALLY PUT MEMORY
015344  012701  077406                 MOV     #77406,R1      ;MANAGEMENT IN A 'PASSIVE'
015350  012702  172300                 MOV     #KIPDR0,R2     ;STATE, THAT IS MAP ALL
015354  012703  000010                 MOV     #10,R3         ;VIRTUAL ADDRESSES ON TO
015360  010122                 64$:    MOV     R1,(R2)+       ;THEMSELVES AS PHYSICAL
015362  077302                         SOB     R3,64$         ;ADDRESSES.
015364  005020                         CLR     (R0)+
015366  012720  000200                 MOV     #200,(R0)+
015372  012720  000400                 MOV     #400,(R0)+
015376  012720  000600                 MOV     #600,(R0)+
015402  012720  001000                 MOV     #1000,(R0)+
015406  012720  001200                 MOV     #1200,(R0)+
015412  012720  001400                 MOV     #1400,(R0)+
015416  012710  177600                 MOV     #177600,(R0)

015422  104424                         SIZE
015424  000000                 AALOAD: .WORD   0              ;ADDRESS OF THE HIGHEST
015426  000000                 AAHIAD: .WORD   0              ;WORD IN MEMORY.
015430  042737  000002  015424         BIC     #2,AALOAD

015436  012700  016234                 MOV     #AATMP2,R0     ;ESTABLISH BITS 9 THROUGH
015442  042700  176003                 BIC     #176003,R0     ;0 TO BE PART OF ALL
015446  010037  016220                 MOV     R0,AAOFST      ;THE TEST ADDRESSES.
015452  005037  016222                 CLR     AAOFST+2

015456  012737  000020  172516         MOV     #20,@#MMR3     ;ENABLE 22-BIT MODE
015464  012737  000001  177572         MOV     #1,@#MMR0      ;ADDRESSING

015472  012737  000030  016210         MOV     #S0M1,AAGS     ;TEST GROUP 0 FIRST, AAGS
015500  005037  016204                 CLR     AAFLG1         ;CONTAINS A PATTERN TO
015504  012737  001400  016212         MOV     #1400,AAERGS   ;BE PUT IN THE CONTROL
015512  012737  004420  016230         MOV     #4420,AAEXER   ;REGISTER. AAERGS CONTAINS
                                                              ;A PATTERN FOR THE MAINT. REG.
015520  012737  000001  016216 AA1:    MOV     #1,AAADR1+2    ;AAADR1 CONTAINS BITS
015526  005037  016214                 CLR     AAADR1         ;10 THROUGH 22 OF
                                                              ;THE TEST ADDRESS.
                                                              ;INITIALIZE IT.
015532  013737  016210  177746         MOV     AAGS,@#CONTRL  ;SELECT THE GROUP BEING
                                                              ;TESTED. MISS THE OTHER
                                                              ;GROUP.
```

```
015540                          AA2:                            ;GET THE TEST ADDRESS
                                                                ;INTO THE AAADR2=AAADR1+AAOFST
                                ;DOUBLE PRECISION ADDITION, UNSIGNED
015540  013737  016214  016224          MOV     AAADR1,AAADR2
015546  013737  016216  016226          MOV     AAADR1+2,AAADR2+2
015554  063737  016220  016224          ADD     AAOFST,AAADR2
015562  005537  016226                  ADC     AAADR2+2
015566  063737  016222  016226          ADD     AAOFST+2,AAADR2+2



                                                                ;SEE IF THIS ADDRESS
                                                                ;IS A REAL MEMORY LOCATION
                                                                ;IF NOT THIS GROUP HAS
                                                                ;BEEN TESTED.

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
015574  023737  016226  015426          CMP     AAADR2+2,AALOAD+2     ;COMPARE THE HIGH ORDER
015602  001006                          BNE     64$             ;PARTS OF AAADR2 AND ARG2.
015604  023737  016224  015424          CMP     AAADR2,AALOAD   ;COMPARE THE LOW ORDER

015612  001002                          BNE     64$             ;PARTS.



015614  000137  015632                  JMP     AA3             ;THEY WERE EQUAL!

015620  103402                  64$:    BLO     65$
015622  000137  016142                  JMP     AA8             ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
015626  000137  015632          65$:    JMP     AA3             ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.


015632  012737  000001  016206  AA3:    MOV     #1,AAFLG2       ;THE ADDRESS IS GOOD! SET
                                                                ;AAFLG2 TO INDICATE AN
                                                                ;ERROR IS BEING FORCED
                                                                ;ON THE LOW BYTE.
                                ;ESTABLISH A VIRTUAL ADDRESS WHICH WILL RELOCATE
                                ;THROUGH KIPAR6 TO THE TEST ADDRESS.
015640  013703  016224                  MOV     AAADR2,R3
015644  013702  016226                  MOV     AAADR2+2,R2
015650  162703  000002                  SUB     #2,R3
015654  005602                          SBC     R2
015656  010300                          MOV     R3,R0
015660  042700  177700                  BIC     #177700,R0      ;R0 CONTAINS THE VIRTUAL
015664  062700  140000                  ADD     #140000,R0      ;ADDRESS.

015670  073227  177772                  ASHC    #-6,R2          ;SET KIPAR6
015674  010337  172354                  MOV     R3,@#KIPAR6

015700  012737  054076  000114          MOV     #SPUR,@#CACHVEC ;RESET VECTOR CACHVEC IN CASE
                                                                ;A PARITY ERROR OCCURS
                                                                ;WHILE SETTING UP THE
                                                                ;INSTRUCTIONS AT THE TEST
                                                                ;ADDRESS.
```

```
                                                                  ;PUT THE INSTRUCTIONS AT
                                                                  ;THE TEST ADDRESS
015706 012710 010112                  MOV     #010112,(R0)        ;010112 = 'MOV R4,(R2)'
015712 012760 005012 000002           MOV     #005012,2(R0)       ;005012 = 'CLR (R2)'
015720 012760 000207 000004           MOV     #000207,4(R0)       ;000207 = 'RTS PC'

015726 005760 000002                  TST     2(R0)               ;MAKE THE TEST ADDRESS
015732 005760 000002                  TST     2(R0)               ;A HIT IN THE GROUP
015736 032737 000010 177752           BIT     #10,@#HITMIS        ;BEING TESTED!
015744 001016                         BNE     AA4

015746 012737 015774 001634           MOV     #1$,$TMP3           ;IF UNABLE TO GET A GIT
015754 013737 016204 001630           MOV     AAFLG1,$TMP1        ;REPORT ERROR!
015762 010037 001632                  MOV     R0,$TMP2
015766 062737 000002 001632           ADD     #2,$TMP2
015774 104001                 1$:     ERROR   1
015776 000137 016124                  JMP     AA7                 ;GO TO NEXT TEST ADDRESS.

016002                        AA4:                                ;THE TEST ADDRESS IS NOW
                                                                  ;A HIT IN THE GROUP
016002 012737 016240 000114           MOV     #AAERR1,@#CACHVEC      ;BEING TESTED. NOW RESET
                                                                  ;CACHVEC TO GO TO THE EXPECTED
                                                                  ;ERROR HANDLER
016010 012702 177750                  MOV     #MAINT,R2           ;SET R2
016014 013704 016212                  MOV     AAERGS,R4           ;SET R4 WHICH WILL BE
016020 042704 005000                  BIC     #5000,R4            ;LOADED INTO THE MAINT.
                                                                  ;REG SO AS TO FORCE
                                                                  ;A LOW BYTE ADDRESS
                                                                  ;MEMORY PARITY ERROR
                                                                  ;IN THE GROUP BEING
                                                                  ;TESTED.
016024 000240                         NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
016026 004710                         JSR     PC,(R0)             ;GO TO THE TEST
                                                                  ;ADDRESS!

016030                        AA5:                                ;RETURN,RTS PC, BACK TO
016030 013737 016204 001632           MOV     AAFLG1,$TMP2        ;HERE IF THE TEST FAILED
016036 013737 016224 001634           MOV     AAADR2,$TMP3        ;TO FORCE AN ERROR AT
016044 013737 016226 001636           MOV     AAADR2+2,$TMP4      ;THE TEST ADDRESS'S LOW
016052 104136                 1$:     ERROR   136                 ;BYTE. REPORT THE FAILURE!

016054                        AA6:                                ;TRY TO DO THE SAME
                                                                  ;THING NOW ONLY FORCE THE
                                                                  ;ERROR ON THE ADDRESSES
                                                                  ;HIGH BYTE!
                                                                  ;THE INSTRUCTIONS ARE
                                                                  ;ALREADY AT THE TEST
016054 012737 000002 016206           MOV     #2,AAFLG2           ;ADDRESS. BUT MAKE SURE
016062 005760 000002                  TST     2(R0)               ;IT IS STILL A HIT!
016066 013704 016212                  MOV     AAERGS,R4           ;SET R4 TO FORCE THE
016072 042704 002400                  BIC     #2400,R4            ;ERROR ON THE HIGH BYTE.
016076 004710                         JSR     PC,(R0)             ;GO DO THE TEST!

016100                        AA16:                               ;RETURN,RTS PC, BACK TO HERE
016100 013737 016204 001632           MOV     AAFLG1,$TMP2        ;IF THE TEST FAILED
016106 013737 016224 001634           MOV     AAADR2,$TMP3        ;IN TRYING TO FORCE A
```

```
016114  013737  016226  001636                  MOV     AAADR2+2,$TMP4   ;ERROR ON THE HIGH BYTE
016122  104137                          1$:     ERROR   137              ;IN THE ADDRESS MEMORY

016124  062737  002000  016214          AA7:    ADD     #2000,AAADR1     ;INCREMENT BITS 21 THROUGH
016132  005537  016216                          ADC     AAADR1+2         ;10 OF THE TEST ADDRESS
016136  000137  015540                          JMP     AA2              ;AND GO TEST THIS NEW
                                                                         ;TEST ADDRESS!
016142  005737  016204                  AA8:    TST     AAFLG1           ;SEE IF BOTH GROUPS HAVE
016146  001111                                  BNE     AADONE           ;BEEN TESTED. IF NOT, GO
016150  012737  004440  016230                  MOV     #4440,AAEXER     ;BACK TO AA1 TO TEST
016156  012737  000044  016210                  MOV     #S1M0,AAGS       ;GROUP ONE, OTHERWISE DONE!
016164  012737  000001  016204                  MOV     #1,AAFLG1
016172  012737  006000  016212                  MOV     #6000,AAERGS
016200  000137  015520                          JMP     AA1

016204  000000                          AAFLG1: .WORD   0                ;A FLAG WHICH INDICATES
                                                                         ;WHICH GROUP IS BEING TESTED
                                                                         ;1 OR 0
016206  000000                          AAFLG2: .WORD   0                ;A FLAG WHICH INDICATES
                                                                         ;WHETHER THE LOW BYTE (1)
                                                                         ;THE HIGH BYTE (2) IS
                                                                         ;BEING TESTED.
016210  000000                          AAGS:   .WORD   0                ;A PATTERN FOR THE CONTROL
                                                                         ;REGISTER.
016212  000000                          AAERGS: .WORD   0                ;PATTERN FOR THE MAINT. REG.
016214  000000                          AAADR1: .WORD   0                ;BITS 21 THROUGH 10 OF
016216  000000                                  .WORD   0                ;THE TEST ADDRESS.
016220  000000                          AAOFST: .WORD   0                ;BITS 9 THROUGH 0 OF
016222  000000                                  .WORD   0                ;THE TEST ADDRESS.
016224  000000                          AAADR2: .WORD   0                ;THE TEST ADDRESS
016226  000000                                  .WORD   0                ;'AAADR2 = AAADR1+AAOFST'
016230  000000                          AAEXER: .WORD   0                ;EXPECTED ERROR REGISTER
016232  000000                          AATMP1: .WORD   0                ;THESE ADDRESSES ARE
016234  000000                          AATMP2: .WORD   0                ;USED TO DETERMINE AAOFST.
016236  000000                                  .WORD   0

016240  016666  000002  000004          AAERR1: MOV     2(SP),4(SP)      ;RESET THE STACK. RECALL THAT THE
016246  012616                                  MOV     (SP)+,(SP)                ;TEST ROUTINE WAS JSR'ED TO AND
                                                                         ;A PARITY ERROR TRAP BROUGHT CONTROL
                                                                         ;BACK!!
016250  023737  016230  177744                  CMP     AAEXER,@#MEMERR  ;MAKE SURE THE ERROR
016256  001405                                  BEQ     1$               ;WHICH OCCURRED WAS
016260  012737  054076  000114                  MOV     #SPUR,@#CACHVEC  ;THE EXPECTED ERROR AT
016266  000137  054076                          JMP     SPUR             ;THE EXPECTED ADDRESS,
                                                                         ;IF NOT GO TO THE
                                                                         ;SPURIOUS ERROR HANDLER,
                                                                         ;SPUR!
016272                                  1$:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
016272  023737  016226  177742                  CMP     AAADR2+2,LOADRS+2         ;COMPARE THE HIGH ORDER
016300  001006                                  BNE     64$              ;PARTS OF AAADR2 AND ARG2.
016302  023737  016224  177740                  CMP     AAADR2,LOADRS    ;COMPARE THE LOW ORDER

016310  001002                                  BNE     64$              ;PARTS.

```

```


016312  000137  016330                       JMP     AAERR2              ;THEY WERE EQUAL!

016316  103402                       64$:    BLO     65$
016320  000137  054076                       JMP     SPUR                ;THE FIRST ADDRESS IS LARGER
                                                                         ;THAN THE SECOND!
016324  000137  054076               65$:    JMP     SPUR                ;THE FIRST IS LESS THAN THE
                                                                         ;SECOND.


016330  012737  177777  177744  AAERR2: MOV     #-1,@#MEMERR        ;IF EVERYTHING WAS
                                                                                      ;CORRECT, CLR THE ERROR
016336  022626                               CMP     (SP)+,(SP)+         ;REGISTER RESET THE
                                                                         ;STACK AND CONTINUE
016340  023727  016206  000002               CMP     AAFLG2,#2           ;TESTING
016346  001002                               BNE     1$
016350  000137  016124                       JMP     AA7                 ;TEST THE NEXT ADDRESS
016354  023727  016206  000001  1$:          CMP     AAFLG2,#1
016362  001002                               BNE     2$
016364  000137  016054                       JMP     AA6                 ;TEST THE HIGH BYTE OF THIS ADDRESS
016370  000000                       2$:     HALT                        ;???HOW DID WE GET HERE?

016372  104416                       AADONE: RSET                        ;DONE!
```

```

                                        ;;*************************************************************
                                        ;*TEST 16       CACHE ADDRESS MEMORY DUAL ADDRESS TEST, UPWARD
                                        ;*
                                        ;*THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS
                                        ;*MEMORY.  AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS'
                                        ;*ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP
                                        ;*BEING TESTED.  OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS
                                        ;*TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY
                                        ;*ON THE SYSTEM.  NOTE THAT HERE THE WORD 'TAG' REFERS
                                        ;*TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21,
                                        ;*WHICH ARE STORED IN THE CACHE ADDRESS MEMORY.  HERE
                                        ;*THE ADDRESS MEMORY IS WRITTEN IN THE UPWARD DIRECTION,
                                        ;*THAT IS 'TAG' 1 IS WRITTEN FIRST, 'TAG' 2 SECOND ETC.
                                        ;*THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED
                                        ;* TO SEE IF IT IS A HIT, THUS MAKING SURE NO
                                        ;*'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER
                                        ;*'TAG'.  NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL
                                        ;*ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD
                                        ;*INVOLVE WRITTING THE 'TAGS' IN THE DOWNWARD DIRECTION
                                        ;*AS WELL AS THE UPWARD DIRECTION.  THE DOWNWARD
                                        ;*WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND
                                        ;*IN TST17.
                                        ;*
                                        ;;*************************************************************
016374  000004                          TST16:  SCOPE
016376  012737  000002  001676                  MOV     #2,$TIMES          ;;DO 2 ITERATIONS
        000016                          UU=$TN-1
016404                                  UU0:
                                                                           ;SET THE SKAD REGISTER
016404  012737  020016  054230                  MOV     #TST17,SKAD        ;IN CASE THE TEST ABORTS.

016412  012737  054076  000114                  MOV     #SPUR,@#CACHVEC ;AT FIRST EXPECT NO ERRORS
016420  113737  001502  001626                  MOVB    $TSTNM,$TMPO
016426  005037  017504                          CLR     UUFLG3             ;ERROR FLAG.
016432  104422                                  MMSKIP

016434  104424                                  SIZE
016436  000000                          UULOAD: .WORD   0                  ;ADDRESS OF THE HIGHEST WORD
016440  000000                          UUHIAD: .WORD   0                  ;IN MEMORY

016442  005037  017500                          CLR     UUFLG1             ;TEST GROUP 0 FIRST.
016446  012737  000034  017522                  MOV     #S0M0M1,UUGS
016454  012737  000054  017524                  MOV     #S1M0M1,UUGM

016462  005037  017502                  UU1:    CLR     UUFLG2             ;CLEAR THE PROGRESS FLAG.
016466  012700  016404                          MOV     #UU0,R0            ;MAKE THIS CODE HITS, IN
016472  012701  001000                          MOV     #1000,R1           ;THE GROUP NOT BEING TESTED.

016476  013737  017522  177746          UU2:    MOV     UUGS,@#CONTRL
016504  005760  002000                          TST     2000(R0)
016510  013737  017524  177746                  MOV     UUGM,@#CONTRL
016516  005720                                  TST     (R0)+
016520  077112                                  SOB     R1,UU2

016522  013701  017522                          MOV     UUGS,R1            ;SELECT THE GROUP BEING TESTED.
```

```
016526  042701  177717                  BIC     #177717,R1
016532  010137  177746                  MOV     R1,@#CONTRL


016536  012700  172340                  MOV     #KIPAR0,R0          ;INITIALLY PUT MEMORY
016542  012701  077406                  MOV     #77406,R1           ;MANAGEMENT IN A 'PASSIVE'
016546  012702  172300                  MOV     #KIPDR0,R2          ;STATE, THAT IS MAP ALL
016552  012703  000010                  MOV     #10,R3              ;VIRTUAL ADDRESSES ON TO
016556  010122                  64$:    MOV     R1,(R2)+            ;THEMSELVES AS PHYSICAL
016560  077302                          SOB     R3,64$              ;ADDRESSES.
016562  005020                          CLR     (R0)+
016564  012720  000200                  MOV     #200,(R0)+
016570  012720  000400                  MOV     #400,(R0)+
016574  012720  000600                  MOV     #600,(R0)+
016600  012720  001000                  MOV     #1000,(R0)+
016604  012720  001200                  MOV     #1200,(R0)+
016610  012720  001400                  MOV     #1400,(R0)+
016614  012710  177600                  MOV     #177600,(R0)

016620  012737  000020  172516          MOV     #20,@#MMR3          ;TURN ON MEMORY MANAGEMENT.
016626  012737  000001  177572          MOV     #1,@#MMR0

016634  005037  017512                  CLR     UUADR2              ;INITIALIZE THE ADDRESSES.
016640  005037  017514                  CLR     UUADR2+2
016644  012737  140000  017506          MOV     #140000,UUADR1
016652  005037  017510                  CLR     UUADR1+2
016656  012701  000400                  MOV     #400,R1             ;COUNTER.
016662  012737  017530  000114          MOV     #UUERR1,@#CACHVEC
016670  012737  000001  017502          MOV     #1,UUFLG2           ;KEEP TRACK OF TEST PROGRESS.
016676                          UU3:
                                ;DOUBLE PRECISION ADDITION, UNSIGNED
016676  013737  017506  017516          MOV     UUADR1,UUADR3
016704  013737  017510  017520          MOV     UUADR1+2,UUADR3+2
016712  063737  017512  017516          ADD     UUADR2,UUADR3
016720  005537  017520                  ADC     UUADR3+2
016724  063737  017514  017520          ADD     UUADR2+2,UUADR3+2




016732                          UU4:


                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
016732  023737  017520  016440          CMP     UUADR3+2,UULOAD+2       ;COMPARE THE HIGH ORDER
016740  001006                          BNE     64$                 ;PARTS OF UUADR3 AND ARG2.
016742  023737  017516  016436          CMP     UUADR3,UULOAD       ;COMPARE THE LOW ORDER

016750  001002                          BNE     64$                 ;PARTS.



016752  000137  017004                  JMP     UU6                 ;THEY WERE EQUAL!

016756  103402                  64$:    BLO     65$
016760  000137  016770                  JMP     UU5                 ;THE FIRST ADDRESS IS LARGER
                                                                    ;THAN THE SECOND!
```

```
016764  000137  017004          65$:    JMP     UU6                     ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.


016770  012737  140000  017506  UU5:    MOV     #140000,UUADR1          ;RESET TO GET VALID ADDRESS.
016776  005037  017510                  CLR     UUADR1+2
017002  000735                          BR      UU3

017004  012702  017516          UU6:    MOV     #UUADR3,R2

017010  011203                          MOV     (R2),R3                 ;GET THE PHYSICAL ADDRESS POINTED
017012  042703  177700                  BIC     #177700,R3              ;TO BY R2 AND ESTABLISH
017016  011205                          MOV     (R2),R5  ;A VIRTUAL ADDRESS WHICH
017020  016204  000002                  MOV     2(R2),R4                ;WILL RELOCATE THROUGH
017024  073427  177772                  ASHC    #-6,R4                  ;KIPAR6. SETUP KIPAR6 AND
017030  010537  172354                  MOV     R5,@#KIPAR6             ;LEAVE THE VIRTUAL ADDRESS
017034  062703  140000                  ADD     #140000,R3              ;IN R3.


017040  005713                          TST     (R3)                    ;GET A HIT AT THE TEST
017042  005713                          TST     (R3)                    ;ADDRESS.

017044  032737  000010  177752          BIT     #10,@#HITMIS
017052  001012                          BNE     UU7
017054  013737  017500  001632          MOV     UUFLG1,$TMP2
017062  013737  017516  001634          MOV     UUADR3,$TMP3
017070  013737  017520  001636          MOV     UUADR3+2,$TMP4
017076  104041                  1$:     ERROR   41
017100  062737  002000  017506  UU7:    ADD     #2000,UUADR1
017106  005537  017510                  ADC     UUADR1+2
017112  062737  000004  017512          ADD     #4,UUADR2               ;LOOP TO WRITE NEXT ADDRESS
017120  005301                          DEC     R1
017122  001402                          BEQ     1$
017124  000137  016676                  JMP     UU3
017130  012737  000002  017502  1$:     MOV     #2,UUFLG2

017136  013700  017524                  MOV     UUGM,R0                 ;FROM NOW ON SELECT THE
017142  042700  177717                  BIC     #177717,R0              ;GROUP NOT BEING TESTED.
017146  010037  177746                  MOV     R0,@#CONTRL

017152  005037  017512          UU8:    CLR     UUADR2                  ;NOW RE-GENERATE ALL THE
017156  005037  017514                  CLR     UUADR2+2                ;ADDRESS WHICH WERE MADE
017162  012737  140000  017506          MOV     #140000,UUADR1          ;HITS, ABOVE, AND MAKE SURE
017170  005037  017510                  CLR     UUADR1+2                ;THEY ARE STILL HITS.
017174  012701  000400                  MOV     #400,R1
017200  012737  000003  017502          MOV     #3,UUFLG2
017206                          UU9:

                                ;DOUBLE PRECISION ADDITION, UNSIGNED
017206  013737  017506  017516          MOV     UUADR1,UUADR3
017214  013737  017510  017520          MOV     UUADR1+2,UUADR3+2
017222  063737  017512  017516          ADD     UUADR2,UUADR3
017230  005537  017520                  ADC     UUADR3+2
017234  063737  017514  017520          ADD     UUADR2+2,UUADR3+2
```

```

017242                                  UU10:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
017242  023737  017520  016440                  CMP     UUADR3+2,UULOAD+2       ;COMPARE THE HIGH ORDER
017250  001006                                  BNE     64$                     ;PARTS OF UUADR3 AND ARG2.
017252  023737  017516  016436                  CMP     UUADR3,UULOAD           ;COMPARE THE LOW ORDER

017260  001002                                  BNE     64$                     ;PARTS.



017262  000137  017314                          JMP     UU12                    ;THEY WERE EQUAL!

017266  103402                          64$:    BLO     65$
017270  000137  017300                          JMP     UU11                    ;THE FIRST ADDRESS IS LARGER
                                                                                ;THAN THE SECOND!
017274  000137  017314                  65$:    JMP     UU12                    ;THE FIRST IS LESS THAN THE
                                                                                ;SECOND.


017300  012737  140000  017506          UU11:   MOV     #140000,UUADR1          ;RESET TO GET A VALID ADDRESS.
017306  005037  017510                          CLR     UUADR1+2
017312  000735                                  BR      UU9

017314  012702  017516                  UU12:   MOV     #UUADR3,R2

017320  011203                                  MOV     (R2),R3                 ;GET THE PHYSICAL ADDRESS POINTED
017322  042703  177700                          BIC     #177700,R3              ;TO BY R2 AND ESTABLISH
017326  011205                                  MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
017330  016204  000002                          MOV     2(R2),R4                ;WILL RELOCATE THROUGH
017334  073427  177772                          ASHC    #-6,R4                  ;KIPAR6. SETUP KIPAR6 AND
017340  010537  172354                          MOV     R5,@#KIPAR6             ;LEAVE THE VIRTUAL ADDRESS
017344  062703  140000                          ADD     #140000,R3              ;IN R3.


017350  005713                                  TST     (R3)    ;STILL A HIT?
017352  032737  000010  177752                  BIT     #10,@#HITMIS
017360  001012                                  BNE     UU13
                                                                                ;NOT A HIT, A DUAL ADDRESSING
017362  013737  017500  001632                  MOV     UUFLG1,$TMP2            ;PROBLEM?
017370  013737  017516  001634                  MOV     UUADR3,$TMP3
017376  013737  017520  001636                  MOV     UUADR3+2,$TMP4
017404  104042                          1$:     ERROR   42

017406  062737  002000  017506          UU13:   ADD     #2000,UUADR1
017414  005537  017510                          ADC     UUADR1+2
017420  062737  000004  017512                  ADD     #4,UUADR2               ;LOOP TO READ NEXT ADDRESS
017426  005301                                  DEC     R1
017430  001402                                  BEQ     1$
017432  000137  017206                          JMP     UU9
017436  012737  000004  017502          1$:     MOV     #4,UUFLG2
017444  005737  017500                  UU14:   TST     UUFLG1  ;TESTED BOTH GROUPS?
017450  001161                                  BNE     UUDONE                  ;YES.
017452  012737  000001  017500                  MOV     #1,UUFLG1               ;NO, GO TEST GROUP 1.
017460  012737  000054  017522                  MOV     #S1MOM1,UUGS
```

```
017466  012737  000034  017524            MOV     #SOMOM1,UUGM
017474  000137  016462                    JMP     UU1

017500  000000                  UUFLG1:   .WORD   0                 ;FLAG WHICH DESIGNATES
                                                                    ;WHICH GROUP IS BEING TESTED,
                                                                    ;1 OR 0.
017502  000000                  UUFLG2:   .WORD   0                 ;DESIGNATES HOW FAR THE
                                                                    ;TEST HAS PROGRESSED.
017504  000000                  UUFLG3:   .WORD   0                 ;ERROR DURING TEST UUFLG2=4
                                                                    ;PHASE.
017506  000000                  UUADR1:   .WORD   0                 ;ADDRESS WRITTEN INTO ADDRESS
017510  000000                            .WORD   0                 ;MEMORY LOCATION
017512  000000                  UUADR2:   .WORD   0                 ;ADDRESS MEMORY LOCATION
017514  000000                            .WORD   0                 ;BEING TESTED
017516  000000                  UUADR3:   .WORD   0                 ;TEST ADDRESS:UUADR3=UUADR1+UUADR2
017520  000000                            .WORD   0

017522  000000                  UUGS:     .WORD   0                 ;PATTERNS FOR THE CACHE CONTROL
017524  000000                  UUGM:     .WORD   0                 ;REGISTER.
017526  000000                  UUTMP:    .WORD   0

017530  032737  000060  177744  UUERR1:   BIT     #60,@#MEMERR      ;WAS THE ERROR A CACHE ADDRESS
017536  001002                            BNE     UUERR2            ;MEMORY PARITY ERROR
017540  000137  054076                    JMP     SPUR

017544                          UUERR2:                             ;REPORT ERROR.
017544  012637  001632                    MOV     (SP)+,$TMP2
017550  005726                            TST     (SP)+
017552  013737  017500  001634            MOV     UUFLG1,$TMP3
017560  013737  177744  001636            MOV     @#MEMERR,$TMP4
017566  013737  017516  001640            MOV     UUADR3,$TMP5
017574  013737  017520  001642            MOV     UUADR3+2,$TMP6
017602  013737  177740  001644            MOV     @#LOADRS,$TMP7
017610  013737  177742  001646            MOV     @#HIADRS,$TMP10
017616  104043                  1$:       ERROR   43

017620  042737  177717  001636            BIC     #177717,$TMP4     ;TRY TO GET THE BAD ADDRESS
017626  013737  177746  017526            MOV     @#CONTRL,UUTMP    ;OUT OF THE ADDRESS MEMORY.
017634  012737  017664  000114            MOV     #UUERR3,@#CACHVEC
017642  013705  177740                    MOV     @#LOADRS,R5
017646  042705  176001                    BIC     #176001,R5
017652  013737  001636  177746            MOV     $TMP4,@#CONTRL
017660  005715                            TST     (R5)
017662  000401                            BR      UUERR4
017664  022626                  UUERR3:   CMP     (SP)+,(SP)+
017666  012737  177777  177744  UUERR4:   MOV     #-1,@#MEMERR

017674  013737  017526  177746            MOV     UUTMP,@#CONTRL    ;RESET THE CONTROL REGISTER.
017702  012737  017530  000114            MOV     #UUERR1,@#CACHVEC

017710  023727  017502  000001            CMP     UUFLG2,#1         ;RETURN, USING UUFLG2 TO
017716  001002                            BNE     1$                ;DECIDE WHERE.
017720  000137  017100                    JMP     UU7
017724  023727  017502  000002  1$:       CMP     UUFLG2,#2
017732  001002                            BNE     2$
017734  000137  017152                    JMP     UU8
```

```
017740  023727  017502  000003  2$:     CMP     UUFLG2,#3
017746  001002                          BNE     3$
017750  000137  017406                  JMP     UU13
017754  023727  017502  000004  3$:     CMP     UUFLG2,#4
017762  001007                          BNE     4$
017764  005737  017504                  TST     UUFLG3
017770  001011                          BNE     UUDONE
017772  005337  017504                  DEC     UUFLG3
017776  000137  017444                  JMP     UU14

020002  005737  017502          4$:     TST     UUFLG2
020006  001002                          BNE     UUDONE          ;??HALT???
020010  000137  016462                  JMP     UU1

020014  104416                  UUDONE:RSET                     ;DONE!



                                ;;*******************************************************************
                                ;*TEST 17        CACHE ADDRESS MEMORY DUAL ADDRESS TEST, DOWNWARD
                                ;*
                                ;*THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS
                                ;*MEMORY.  AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS'
                                ;*ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP
                                ;*BEING TESTED.  OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS
                                ;*TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY
                                ;*ON THE SYSTEM.  NOTE THAT HERE THE WORD 'TAG' REFERS
                                ;*TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21,
                                ;*WHICH ARE STORED IN THE CACHE ADDRESS MEMORY.  HERE
                                ;*THE ADDRESS MEMORY IS WRITTEN IN THE DOWNWARD DIRECTION,
                                ;*THAT IS 'TAG' 256 IS WRITTEN FIRST, 'TAG' 255 SECOND ETC.
                                ;*THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED
                                ;* TO SEE IF IT IS A HIT, THUS MAKING SURE NO
                                ;*'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER
                                ;*'TAG'.  NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL
                                ;*ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD
                                ;*INVOLVE WRITTING THE 'TAGS' IN THE UPWARD DIRECTION
                                ;*AS WELL AS THE DOWNWARD DIRECTION.  THE UPWARD
                                ;*WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND
                                ;*IN TST16.
                                ;*
                                ;;*******************************************************************
020016  000004                  TST17:  SCOPE
020020  012737  000002  001676          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
        000017                  VV=$TN-1
020026                          VV0:
                                                                ;SET THE SKAD REGISTER
020026  012737  021444  054230          MOV     #TST20,SKAD     ;IN CASE THE TEST ABORTS.

020034  012737  054076  000114          MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
020042  113737  001502  001626          MOVB    $TSTNM,$TMP0

020050  005037  021132                  CLR     VVFLG3          ;CLEAR THE ERROR FLAG.

020054  104422                          MMSKIP

020056  104424                          SIZE
```

```
020060 000000               VVLOAD: .WORD   0               ;ADDRESS OF THE HIGHEST
020062 000000               VVHIAD: .WORD   0               ;WORD IN MEMORY.

020064 005037 021126                CLR     VVFLG1          ;TEST GROUP 0 FIRST
020070 012737 000034 021150         MOV     #S0M0M1,VVGS
020076 012737 000054 021152         MOV     #S1M0M1,VVGM

020104 005037 021130        VV1:    CLR     VVFLG2          ;CLEAR THE PROGRESS FLAG
020110 012700 020026                MOV     #VV0,R0         ;MAKE THIS CODE HITS IN
020114 012701 001000                MOV     #1000,R1        ;THE GROUP NOT BEING
                                                            ;TESTED.
020120 013737 021150 177746 VV2:    MOV     VVGS,@#CONTRL
020126 005760 002000                TST     2000(R0)
020132 013737 021152 177746         MOV     VVGM,@#CONTRL
020140 005720                       TST     (R0)+
020142 077112                       SOB     R1,VV2

020144 013700 021150                MOV     VVGS,R0         ;FROM NOW ON SELECT
020150 042700 177717                BIC     #177717,R0      ;THE GROUP BEING TESTED.
020154 010037 177746                MOV     R0,@#CONTRL


020160 012700 172340                MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
020164 012701 077406                MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
020170 012702 172300                MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
020174 012703 000010                MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
020200 010122               64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSI(AL
020202 077302                       SOB     R3,64$          ;ADDRESSES.
020204 005020                       CLR     (R0)+
020206 012720 000200                MOV     #200,(R0)+
020212 012720 000400                MOV     #400,(R0)+
020216 012720 000600                MOV     #600,(R0)+
020222 012720 001000                MOV     #1000,(R0)+
020226 012720 001200                MOV     #1200,(R0)+
020232 012720 001400                MOV     #1400,(R0)+
020236 012710 177600                MOV     #177600,(R0)

020242 012737 000020 172516         MOV     #20,@#MMR3      ;TURN ON MEMORY MANAGEMENT.
020250 012737 000001 177572         MOV     #1,@#MMR0

020256 012737 001774 021140         MOV     #1774,VVADR2    ;INITIALIZE THE ADDRESSES
020264 005037 021142                CLR     VVADR2+2
020270 012737 140000 021134         MOV     #140000,VVADR1
020276 005037 021136                CLR     VVADR1+2
020302 012701 000400                MOV     #400,R1         ;A COUNTER.
020306 012737 021156 000114         MOV     #VVERR1,@#CACHVEC         ;EXPECT ERRORS NOW.
020314 012737 000001 021130         MOV     #1,VVFLG2       ;KEEP TRACK OF TEST PROGRESS.

020322                      VV3:
                            ;DOUBLE PRECISION ADDITION, UNSIGNED
020322 013737 021134 021144         MOV     VVADR1,VVADR3
020330 013737 021136 021146         MOV     VVADR1+2,VVADR3+2
020336 063737 021140 021144         ADD     VVADR2,VVADR3
020344 005537 021146                ADC     VVADR3+2
020350 063737 021142 021146         ADD     VVADR2+2,VVADR3+2
                                    .
```

```



020356                              VV4:

                                    ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
020356  023737  021146  020062              CMP     VVADR3+2,VVLOAD+2       ;COMPARE THE HIGH ORDER
020364  001006                              BNE     64$                     ;PARTS OF VVADR3 AND ARG2.
020366  023737  021144  020060              CMP     VVADR3,VVLOAD           ;COMPARE THE LOW ORDER

020374  001002                              BNE     64$                     ;PARTS.



020376  000137  020430                      JMP     VV6                     ;THEY WERE EQUAL!

020402  103402                      64$:    BLO     65$
020404  000137  020414                      JMP     VV5                     ;THE FIRST ADDRESS IS LARGER
                                                                            ;THAN THE SECOND!
020410  000137  020430              65$:    JMP     VV6                     ;THE FIRST IS LESS THAN THE
                                                                            ;SECOND.


020414  012737  140000  021134      VV5:    MOV     #140000,VVADR1          ;RESET TO GET A VALID ADDRESS.
020422  005037  021136                      CLR     VVADR1+2
020426  000735                              BR      VV3

020430  012702  021144              VV6:    MOV     #VVADR3,R2

020434  011203                              MOV     (R2),R3                 ;GET THE PHYSICAL ADDRESS POINTED
020436  042703  177700                      BIC     #177700,R3              ;TO BY R2 AND ESTABLISH
020442  011205                              MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
020444  016204  000002                      MOV     2(R2),R4                ;WILL RELOCATE THROUGH
020450  073427  177772                      ASHC    #-6,R4                  ;KIPAR6. SETUP KIPAR6 AND
020454  010537  172354                      MOV     R5,@#KIPAR6             ;LEAVE THE VIRTUAL ADDRESS
020460  062703  140000                      ADD     #140000,R3              ;IN R3.


020464  005713                              TST     (R3)                    ;GET A HIT AT THE
020466  005713                              TST     (R3)                    ;TEST ADDRESS
020470  032737  000010  177752              BIT     #10,@#HITMIS
020476  001012                              BNE     VV7
                                                                            ;REPORT FAILURE TO GET A HIT.
020500  013737  021126  001632              MOV     VVFLG1,$TMP2
020506  013737  021144  001634              MOV     VVADR3,$TMP3
020514  013737  021146  001636              MOV     VVADR3+2,$TMP4
020522  104041                      1$:     ERROR   41

020524  062737  002000  021134      VV7:    ADD     #2000,VVADR1
020532  005537  021136                      ADC     VVADR1+2
020536  062737  177774  021140              ADD     #-4,VVADR2              ;LOOP TO WRITE NEXT ADDRESS
020544  005301                              DEC     R1
020546  001402                              BEQ     1$
020550  000137  020322                      JMP     VV3
020554  012737  000002  021130      1$:     MOV     #2,VVFLG2

```

```
020562  013700  021152                    MOV     VVGM,R0            ;FROM NOW ON SELECT
020566  042700  177717                    BIC     #177717,R0         ;THE GROUP NOT BEING
020572  010037  177746                    MOV     R0,@#CONTRL        ;TESTED.

020576  012737  001774  021140  VV8:      MOV     #1774,VVADR2       ;NOW RE-GENERATE ALL THE
020604  005037  021142                    CLR     VVADR2+2           ;ADDRESSES MADE HITS IN
020610  012737  140000  021134            MOV     #140000,VVADR1     ;THE ABOVE PORTION OF
020616  005037  021136                    CLR     VVADR1+2           ;THE TEST, AND MAKE SURE
020622  012701  000400                    MOV     #400,R1            ;THEY ARE STILL HITS.
020626  012737  000003  021130            MOV     #3,VVFLG2
020634                          VV9:
                                ;DOUBLE PRECISION ADDITION, UNSIGNED
020634  013737  021134  021144            MOV     VVADR1,VVADR3
020642  013737  021136  021146            MOV     VVADR1+2,VVADR3+2
020650  063737  021140  021144            ADD     VVADR2,VVADR3
020656  005537  021146                    ADC     VVADR3+2
020662  063737  021142  021146            ADD     VVADR2+2,VVADR3+2




020670                          VV10:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
020670  023737  021146  020062            CMP     VVADR3+2,VVLOAD+2        ;COMPARE THE HIGH ORDER
020676  001006                            BNE     64$                ;PARTS OF VVADR3 AND ARG2.
020700  023737  021144  020060            CMP     VVADR3,VVLOAD      ;COMPARE THE LOW ORDER

020706  001002                            BNE     64$                ;PARTS.



020710  000137  020742                    JMP     VV12               ;THEY WERE EQUAL!

020714  103402                  64$:      BLO     65$
020716  000137  020726                    JMP     VV11               ;THE FIRST ADDRESS IS LARGER
                                                                     ;THAN THE SECOND!
020722  000137  020742          65$:      JMP     VV12               ;THE FIRST IS LESS THAN THE
                                                                     ;SECOND.


020726  012737  140000  021134  VV11:     MOV     #140000,VVADR1     ;RESET TO CREATE A VALID
020734  005037  021136                    CLR     VVADR1+2           ;ADDRESS
020740  000735                            BR      VV9

020742  012702  021144          VV12:     MOV     #VVADR3,R2

020746  011203                            MOV     (R2),R3            ;GET THE PHYSICAL ADDRESS POINTED
020750  042703  177700                    BIC     #177700,R3         ;TO BY R2 AND ESTABLISH
020754  011205                            MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
020756  016204  000002                    MOV     2(R2),R4           ;WILL RELOCATE THROUGH
020762  073427  177772                    ASHC    #-6,R4             ;KIPAR6. SETUP KIPAR6 AND
020766  010537  172354                    MOV     R5,@#KIPAR6        ;LEAVE THE VIRTUAL ADDRESS
020772  062703  140000                    ADD     #140000,R3         ;IN R3.
```

```
021776  005713                           TST     (R3)             ;STILL A HIT?
021000  032737  000010  177752           BIT     #10,@#HITMIS
021006  001012                           BNE     VV13
                                                                  ;REPORT ERROR.
021010  013737  021126  001632           MOV     VVFLG1,$TMP2
021016  013737  021144  001634           MOV     VVADR3,$TMP3
021024  013737  021146  001636           MOV     VVADR3+2,$TMP4
021032  104042                   1$:     ERROR   42

021034  062737  002000  021134  VV13:    ADD     #2000,VVADR1
021042  005537  021136                   ADC     VVADR1+2
021046  062737  177774  021140           ADD     #-4,VVADR2
021054  005301                           DEC     R1
021056  001402                           BEQ     1$
021060  000137  020634                   JMP     VV9
021064  012737  000004  021130  1$:      MOV     #4,VVFLG2
021072  005737  021126          VV14:    TST     VVFLG1           ;TESTED BOTH GROUPS?
021076  001161                           BNE     VVDONE           ;YES.
021100  012737  000034  021152           MOV     #S0M0M1,VVGM     ;NO GO TEST GROUP 1.
021106  012737  000054  021150           MOV     #S1M0M1,VVGS
021114  012737  000001  021126           MOV     #1,VVFLG1
021122  000137  020104                   JMP     VV1

021126  000000                  VVFLG1:  .WORD   0                ;0 OR 1, GROUP BEING TESTED.
021130  000000                  VVFLG2:  .WORD   0                ;TEST PROGRESS FLAG.
021132  000000                  VVFLG3:  .WORD   0                ;ERROR FLAG.

021134  000000                  VVADR1:  .WORD   0                ;PATTERN WRITTEN INTO THE ADDRESS
021136  000000                           .WORD   0                ;MEMORY LOCATION.
021140  000000                  VVADR2:  .WORD   0                ;ADDRESS MEMORY LOCATION BEING
021142  000000                           .WORD   0                ;TESTED X 4.
021144  000000                  VVADR3:  .WORD   0                ;TEST ADDRESS.
021146  000000                           .WORD   0                ;VVADR3=VVADR2+VVADR1

021150  000000                  VVGS:    .WORD   0                ;PATTERNS FOR THE CACHE
021152  000000                  VVGM:    .WORD   0                ;CONTROL REGISTER.

021154  000000                  VVTMP:   .WORD   0

021156  032737  000060  177744  VVERR1:  BIT     #60,@#MEMERR     ;WAS THE ERROR THAT CAUSED
021164  001002                           BNE     VVERR2           ;THE TRAP TO HERE A CACHE
021166  000137  054076                   JMP     SPUR             ;ADDRESS MEMORY PARITY ERROR?

021172                          VVERR2:                           ;REPORT ERROR.
021172  012637  001632                   MOV     (SP)+,$TMP2
021176  005726                           TST     (SP)+
021200  013737  021126  001634           MOV     VVFLG1,$TMP3
021206  013737  177744  001636           MOV     @#MEMERR,$TMP4
021214  013737  021144  001640           MOV     VVADR3,$TMP5
021222  013737  021146  001642           MOV     VVADR3+2,$TMP6
021230  013737  177740  001644           MOV     @#LOADRS,$TMP7
021236  013737  177742  001646           MOV     @#HIADRS,$TMP10
021244  104043                   1$:     ERROR   43

021246  042737  177717  001636           BIC     #177717,$TMP4    ;TRY TO GET THE BAD ADDRESS
021254  013737  177746  021154           MOV     @#CONTRL,VVTMP   ;OUT OF THE ADDRESS MEMORY.
```

```
021262  012737  021312  000114                  MOV     #VVERR3,@#CACHVEC
021270  013705  177740                          MOV     @#LOADRS,R5
021274  042705  176001                          BIC     #176001,R5
021300  013737  001636  177746                  MOV     $TMP4,@#CONTRL
021306  005715                                  TST     (R5)
021310  000401                                  BR      VVERR4
021312  022626                          VVERR3: CMP     (SP)+,(SP)+
021314  012737  177777  177744          VVERR4: MOV     #-1,@#MEMERR

021322  013737  021154  177746                  MOV     VVTMP,@#CONTRL  ;RESET THE CONTRL REGISTER
021330  012737  021156  000114                  MOV     #VVERR1,@#CACHVEC
021336  023727  021130  000001                  CMP     VVFLG2,#1       ;RETURN, USING VVFLG2 TO
021344  001002                                  BNE     1$              ;DECIDE WHERE.
021346  000137  020524                          JMP     VV7
021352  023727  021130  000002          1$:     CMP     VVFLG2,#2
021360  001002                                  BNE     2$
021362  000137  020576                          JMP     VV8
021366  023727  021130  000003          2$:     CMP     VVFLG2,#3
021374  001002                                  BNE     3$
021376  000137  021034                          JMP     VV13
021402  023727  021130  000004          3$:     CMP     VVFLG2,#4
021410  001007                                  BNE     4$
021412  005737  021132                          TST     VVFLG3
021416  001011                                  BNE     VVDONE
021420  005337  021132                          DEC     VVFLG3
021424  000137  021072                          JMP     VV14
021430  005737  021130                  4$:     TST     VVFLG2
021434  001002                                  BNE     VVDONE          ;????HALT???
021436  000137  020104                          JMP     VV1

021442  104416                          VVDONE: RSET                            ;DONE!



                                        ;;*******************************************************************
                                        ;*TEST 20       CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ONES TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC.  THIS
                                        ;*IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING
                                        ;*A WRITE.  IT DESIGNATES WHICH BYTES OF THE TWO WORDS OF
                                        ;*DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO
                                        ;*BE WRITTEN.  THIS WILL BE A TEST DOING CPU DATOB REFERENCES TO
                                        ;*THE CACHE.  THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN
                                        ;*OF ZEROES.
                                        ;*
                                        ;;*******************************************************************
021444  000004                          TST20:  SCOPE
021446  012737  000010  001676                  MOV     #10,$TIMES      ;;DO 10 ITERATIONS
        000020                          CC=$TN-1
                                                                        ;SET THE SKAD REGISTER
021454  012737  022230  054230                  MOV     #TST21,SKAD     ;IN CASE THE TEST ABORTS.

021462  113737  001502  001626                  MOVB    $TSTNM,$TMP0
021470  012737  021734  000114                  MOV     #CCERR1,@#CACHVEC

021476  012737  000014  177746                  MOV     #MOM1,@#CONTRL  ;FORCE MISSES

```

```
021504  012700  021730                    MOV     #CCTMP2,R0          ;LOCATE THE TEST SPACE.
021510  042700  000003                    BIC     #3,R0
021514  010001                            MOV     R0,R1
021516  005010                    CC1:    CLR     (R0)                ;TEST MASK 0
021520  005060  000002                    CLR     2(R0)
021524  000240                            NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
021526  112711  000377                    MOVB    #377,(R1)
021532  022710  000377                    CMP     #377,(R0)
021536  001403                            BEQ     CC3
021540  004737  022146            CC2:    JSR     PC,CCERR3
021544  000403                            BR      CC4
021546  005760  000002            CC3:    TST     2(R0)
021552  001372                            BNE     CC2
021554  062701  000001            CC4:    ADD     #1,R1               ;TEST MASK 1.
021560  005010                            CLR     (R0)
021562  005060  000002                    CLR     2(R0)
021566  000240                            NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
021570  112711  000377                    MOVB    #377,(R1)
021574  022710  177400                    CMP     #177400,(R0)
021600  001403                            BEQ     CC6
021602  004737  022146            CC5:    JSR     PC,CCERR3
021606  000403                            BR      CC7
021610  005760  000002            CC6:    TST     2(R0)
021614  001372                            BNE     CC5

021616  062701  000001            CC7:    ADD     #1,R1               ;TEST MASK 2.
021622  005010                            CLR     (R0)
021624  005060  000002                    CLR     2(R0)
021630  000240                            NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
021632  112711  000377                    MOVB    #377,(R1)
021636  022760  000377  000002            CMP     #377,2(R0)
021644  001403                            BEQ     CC9
021646  004737  022146            CC8:    JSR     PC,CCERR3
021652  000402                            BR      CC10
021654  005710                    CC9:    TST     (R0)
021656  001373                            BNE     CC8

021660  062701  000001            CC10:   ADD     #1,R1               ;TEST MASK 3.
021664  005010                            CLR     (R0)
021666  005060  000002                    CLR     2(R0)
021672  000240                            NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
021674  112711  000377                    MOVB    #377,(R1)
021700  022760  177400  000002            CMP     #177400,2(R0)
021706  001403                            BEQ     CC12
021710  004737  022146            CC11:   JSR     PC,CCERR3
021714  000402                            BR      CC13
021716  005710                    CC12:   TST     (R0)
021720  001373                            BNE     CC11

021722  000137  022226            CC13:   JMP     CCDONE

021726  000000                    CCTMP1: .WORD   0
021730  000000                    CCTMP2: .WORD   0                   ;THE TEST AREA.
021732  000000                            .WORD   0
```

```
021734  032737  000002  177744  CCERR1: BIT     #2,@#MEMERR         ;SHOULD BE A MAIN MEMORY
021742  001002                          BNE     1$                  ;ADDRESS AND CONTROL LINE
021744  000137  054076                  JMP     SPUR                ;PARITY ERROR.
021750  020137  177740          1$:     CMP     R1,@#LOADRS         ;ERROR ADDRESS SHOULD BE
021754  001402                          BEQ     CCERR2              ;TEST ADDRESS.
021756  000137  054076                  JMP     SPUR
021762  012637  001642          CCERR2: MOV     (SP)+,$TMP6
021766  005037  001664                  CLR     $TMP17
021772  005726                          TST     (SP)+               ;RESET THE STACK
021774  012737  000044  001666          MOV     #44,$TMP20
022002  013737  177740  001634          MOV     @#LOADRS,$TMP3
022010  013737  177742  001636          MOV     @#HIADRS,$TMP4
022016  013737  177744  001640          MOV     @#MEMERR,$TMP5
022024  010037  001642                  MOV     R0,$TMP6
022030  005037  001644                  CLR     $TMP7
022034  010037  001656                  MOV     R0,$TMP14
022040  062737  000002  001656          ADD     #2,$TMP14
022046  005037  001660                  CLR     $TMP15
022052  011037  001646                  MOV     (R0),$TMP10
022056  016037  000002  001650          MOV     2(R0),$TMP11
022064  010137  001652                  MOV     R1,$TMP12
022070  005037  001654                  CLR     $TMP13
022074  104044                  64$:    ERROR   44
022076  012737  177777  177744          MOV     #-1,@#MEMERR

022104  010002                          MOV     R0,R2
022106  020102                          CMP     R1,R2
022110  001002                          BNE     2$
022112  000137  021554                  JMP     CC4
022116  005202                  2$:     INC     R2
022120  020102                          CMP     R1,R2
022122  001002                          BNE     3$
022124  000137  021616                  JMP     CC7
022130  005202                  3$:     INC     R2
022132  020102                          CMP     R1,R2
022134  001002                          BNE     4$
022136  000137  021660                  JMP     CC10
022142  000137  022226          4$:     JMP     CCDONE


022146  011637  001652          CCERR3: MOV     (SP),$TMP12         ;REPORT FAILURE TO WRITE
                                                                    ;THE CORRECT BYTE
022152  010037  001632                  MOV     R0,$TMP2
022156  005037  001634                  CLR     $TMP3
022162  010037  001636                  MOV     R0,$TMP4
022166  062737  000002  001636          ADD     #2,$TMP4
022174  005037  001640                  CLR     $TMP5
022200  011037  001642                  MOV     (R0),$TMP6
022204  016037  000002  001644          MOV     2(R0),$TMP7
022212  010137  001646                  MOV     R1,$TMP10
022216  005037  001650                  CLR     $TMP11
022222  104046                          ERROR   46
022224  000207                          RTS     PC


022226  104416                  CCDONE: RSET                        ;DONE!
```

```


                                        ;;*******************************************************************
                                        ;*TEST 21         CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ZEROES TEST
                                        ;*
                                        ;*THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC.
                                        ;*HERE CPU DATOB'S WILL MOVE ZEROES INTO A BACKROUND
                                        ;*PATTERN OF ONES.
                                        ;*
                                        ;;*******************************************************************
022230  000004                          TST21:  SCOPE
022232  012737  000010  001676                  MOV     #10,$TIMES          ;;DO 10 ITERATIONS
        000021                          FF=$TN-1
                                                                            ;SET THE SKAD REGISTER
022240  012737  023026  054230                  MOV     #TST22,SKAD         ;IN CASE THE TEST ABORTS.

022246  113737  001502  001626                  MOVB    $TSTNM,$TMPO
022254  012737  022532  000114                  MOV     #FFERR1,@#CACHVEC

022262  012737  000014  177746                  MOV     #MOM1,@#CONTRL      ;FORCE MISSES.

022270  012700  022526                          MOV     #FFTMP2,R0
022274  042700  000003                          BIC     #3,R0
022300  010001                                  MOV     R0,R1

022302  012710  177777                  FF1:    MOV     #-1,(R0)            ;TEST MASK 0
022306  012760  177777  000002                  MOV     #-1,2(R0)
022314  000240                                  NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
022316  105011                                  CLRB    (R1)
022320  022710  177400                          CMP     #177400,(R0)
022324  001403                                  BEQ     FF3
022326  004737  022744                  FF2:    JSR     PC,FFERR3
022332  000404                                  BR      FF4
022334  022760  177777  000002          FF3:    CMP     #-1,2(R0)
022342  001371                                  BNE     FF2

022344  005201                          FF4:    INC     R1                  ;TEST MASK 1.
022346  012710  177777                          MOV     #-1,(R0)
022352  012760  177777  000002                  MOV     #-1,2(R0)
022360  000240                                  NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
022362  105011                                  CLRB    (R1)
022364  022710  000377                          CMP     #377,(R0)
022370  001403                                  BEQ     FF6
022372  004737  022744                  FF5:    JSR     PC,FFERR3
022376  000404                                  BR      FF7
022400  022760  177777  000002          FF6:    CMP     #-1,2(R0)
022406  001371                                  BNE     FF5

022410  005201                          FF7:    INC     R1                  ;TEST MASK 2.
022412  012710  177777                          MOV     #-1,(R0)
022416  012760  177777  000002                  MOV     #-1,2(R0)
022424  000240                                  NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
022426  105011                                  CLRB    (R1)
022430  022760  177400  000002                  CMP     #177400,2(R0)
022436  001403                                  BEQ     FF9
022440  004737  022744                  FF8:    JSR     PC,FFERR3
```

```
022444  000403                            BR      FF10
022446  022710  177777           FF9:     CMP     #-1,(R0)
022452  001372                            BNE     FF8

022454  005201                   FF10:    INC     R1                  ;TEST MASK 3.
022456  012710  177777                    MOV     #-1,(R0)
022462  012760  177777  000002            MOV     #-1,2(R0)
022470  000240                            NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
022472  105011                            CLRB    (R1)
022474  022760  000377  000002            CMP     #377,2(R0)
022502  001403                            BEQ     FF12
022504  004737  022744           FF11:    JSR     PC,FFERR3
022510  000403                            BR      FF13
022512  022710  177777           FF12:    CMP     #-1,(R0)
022516  001372                            BNE     FF11

022520  000137  023024           FF13:    JMP     FFDONE

022524  000000                   FFTMP1:  .WORD   0                   ;TEST AREA.
022526  000000                   FFTMP2:  .WORD   0
022530  000000                            .WORD   0


022532  032737  000002  177744   FFERR1:  BIT     #2,@#MEMERR         ;SHOULD BE A MAIN MEMORY
022540  001002                            BNE     1$                  ;ADDRESS AND CONTROL LINE
022542  000137  054076                    JMP     SPUR                ;PARITY ERROR.
022546  020137  177740           1$:      CMP     R1,@#LOADRS         ;ERROR ADDRESS SHOULD BE
022552  001402                            BEQ     FFERR2              ;TEST ADDRESS.
022554  000137  054076                    JMP     SPUR
022560  012637  001642           FFERR2:  MOV     (SP)+,$TMP6
022564  005037  001664                    CLR     $TMP17
022570  005726                            TST     (SP)+               ;RESET THE STACK
022572  012737  000050  001666            MOV     #50,$TMP20
022600  013737  177740  001634            MOV     @#LOADRS,$TMP3
022606  013737  177742  001636            MOV     @#HIADRS,$TMP4
022614  013737  177744  001640            MOV     @#MEMERR,$TMP5
022622  010037  001642                    MOV     R0,$TMP6
022626  005037  001644                    CLR     $TMP7
022632  010037  001656                    MOV     R0,$TMP14
022636  062737  000002  001656            ADD     #2,$TMP14
022644  005037  001660                    CLR     $TMP15
022650  011037  001646                    MOV     (R0),$TMP10
022654  016037  000002  001650            MOV     2(R0),$TMP11
022662  010137  001652                    MOV     R1,$TMP12
022666  005037  001654                    CLR     $TMP13
022672  104050                   64$:     ERROR   50
022674  012737  177777  177744            MOV     #-1,@#MEMERR

022702  010002                            MOV     R0,R2
022704  020102                            CMP     R1,R2
022706  001002                            BNE     2$
022710  000137  022344                    JMP     FF4
022714  005202                   2$:      INC     R2
022716  020102                            CMP     R1,R2
022720  001002                            BNE     3$
022722  000137  022410                    JMP     FF7
```

```
022726  005202                          3$:     INC     R2
022730  020102                                  CMP     R1,R2
022732  001002                                  BNE     4$
022734  000137  022454                          JMP     FF10
022740  000137  023024                  4$:     JMP     FFDONE          ;HALT????


022744  011637  001652                  FFERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
                                                                        ;THE CORRECT BYTE
022750  010037  001632                          MOV     R0,$TMP2
022754  005037  001634                          CLR     $TMP3
022760  010037  001636                          MOV     R0,$TMP4
022764  062737  000002  001636                  ADD     #2,$TMP4
022772  005037  001640                          CLR     $TMP5
022776  011037  001642                          MOV     (R0),$TMP6
023002  016037  000002  001644                  MOV     2(R0),$TMP7
023010  010137  001646                          MOV     R1,$TMP10
023014  005037  001650                          CLR     $TMP11
023020  104052                                  ERROR   52
023022  000207                                  RTS     PC


023024  104416                          FFDONE: RSET                    ;DONE!

                                        ;;*******************************************************************
                                        ;*TEST 22        CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ONES TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC.  THIS
                                        ;*IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING
                                        ;*A WRITE.  IT DESIGNATES WHICH BYTES OF THE TWO WORDS OF
                                        ;*DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO
                                        ;*BE WRITTEN.  THIS WILL BE A TEST DOING UNIBUS DATOB REFERENCES TO
                                        ;*THE CACHE.  THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN
                                        ;*OF ZEROES.
                                        ;*
                                        ;;*******************************************************************
023026  000004                          TST22:  SCOPE
023030  012737  000010  001676                  MOV     #10,$TIMES      ;;DO 10 ITERATIONS
        000022                          EE=$TN-1
                                                                        ;SET THE SKAD REGISTER
023036  012737  023714  054230                  MOV     #TST23,SKAD     ;IN CASE THE TEST ABORTS.

023044  113737  001502  001626                  MOVB    $TSTNM,$TMP0
023052  104422                                  MMSKIP
023054  012737  023420  000114                  MOV     #EEERR1,@#CACHVEC

023062  012700  172340                          MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                        ;TO RELOCATE EVERYTHING
023066  012702  172300                          MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
023072  012703  000007                          MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
023076  005004                                  CLR     R4              ;BY PASSIVELY IS MEANT
023100  012705  170200                          MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                                        ;RELOCATED TO THEMSELVES.
023104  012722  077406                  64$:    MOV     #77406,(R2)+
023110  010401                                  MOV     R4,R1
023112  072127  000006                          ASH     #6,R1
```

```
023116  010125                                  MOV     R1,(R5)+
023120  005025                                  CLR     (R5)+
023122  010410                                  MOV     R4,(R0)
023124  062720  170000                          ADD     #170000,(R0)+
023130  062704  000200                          ADD     #200,R4
023134  077315                                  SOB     R3,64$
023136  012710  177600                          MOV     #177600,(R0)
023142  012712  077406                          MOV     #77406,(R2)

023146  012737  000060  172516                  MOV     #60,@#MMR3          ;TURN ON MEMORY MANAGEMENT
023154  012737  000001  177572                  MOV     #1,@#MMR0           ;AND THE MAPPING BOX RELOCATION.

023162  012737  000014  177746                  MOV     #MOM1,@#CONTRL      ;FORCE MISSES TO BOTH GROUPS.

023170  012700  023414                          MOV     #EETMP2,R0          ;LOCATE THE TEST SPACE.
023174  042700  000003                          BIC     #3,R0
023200  010001                                  MOV     R0,R1

023202  005010                          EE1:    CLR     (R0)                ;TEST MASK 0
023204  005060  000002                          CLR     2(R0)
023210  000240                                  NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
023212  112711  000377                          MOVB    #377,(R1)
023216  022710  000377                          CMP     #377,(R0)
023222  001403                                  BEQ     EE3
023224  004737  023632                  EE2:    JSR     PC,EEERR3
023230  000403                                  BR      EE4
023232  005760  000002                  EE3:    TST     2(R0)
023236  001372                                  BNE     EE2

023240  062701  000001                  EE4:    ADD     #1,R1
023244  005010                                  CLR     (R0)
023246  005060  000002                          CLR     2(R0)
023252  000240                                  NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
023254  112711  000377                          MOVB    #377,(R1)
023260  022710  177400                          CMP     #177400,(R0)
023264  001403                                  BEQ     EE6
023266  004737  023632                  EE5:    JSR     PC,EEERR3
023272  000403                                  BR      EE7
023274  005760  000002                  EE6:    TST     2(R0)
023300  001372                                  BNE     EE5

023302  062701  000001                  EE7:    ADD     #1,R1
023306  005010                                  CLR     (R0)
023310  005060  000002                          CLR     2(R0)
023314  000240                                  NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
023316  112711  000377                          MOVB    #377,(R1)
023322  022760  000377  000002                  CMP     #377,2(R0)
023330  001403                                  BEQ     EE9
023332  004737  023632                  EE8:    JSR     PC,EEERR3
023336  000402                                  BR      EE10
023340  005710                          EE9:    TST     (R0)
023342  001373                                  BNE     EE8

023344  062701  000001                  EE10:   ADD     #1,R1
023350  005010                                  CLR     (R0)
023352  005060  000002                          CLR     2(R0)
```

```
023356 000240                                   NOP     ;FOR SCOPING WITH AN OSCILLOSCOPE.
023360 112711 000377                            MOVB    #377,(R1)
023364 022760 177400 000002                     CMP     #177400,2(R0)
023372 001403                                   BEQ     EE12
023374 004737 023632                    EE11:   JSR     PC,EEERR3
023400 000402                                   BR      EE13
023402 005710                           EE12:   TST     (R0)
023404 001373                                   BNE     EE11

023406 000137 023712                    EE13:   JMP     EEDONE

023412 000000                           EETMP1: .WORD   0
023414 000000                           EETMP2: .WORD   0
023416 000000                                   .WORD   0


023420 032737 000002 177744             EEERR1: BIT     #2,@#MEMERR             ;SHOULD BE A MAIN MEMORY
023426 001002                                   BNE     1$                      ;ADDRESS AND CONTROL LINE
023430 000137 054076                            JMP     SPUR                    ;PARITY ERROR.
023434 020137 177740                    1$:     CMP     R1,@#LOADRS             ;ERROR ADDRESS SHOULD BE
023440 001402                                   BEQ     EEERR2                  ;TEST ADDRESS.
023442 000137 054076                            JMP     SPUR
023446 012637 001642                    EEERR2: MOV     (SP)+,$TMP6
023452 005037 001664                            CLR     $TMP17
023456 005726                                   TST     (SP)+                   ;RESET THE STACK
023460 012737 000045 001666                     MOV     #45,$TMP20
023466 013737 177740 001634                     MOV     @#LOADRS,$TMP3
023474 013737 177742 001636                     MOV     @#HIADRS,$TMP4
023502 013737 177744 001640                     MOV     @#MEMERR,$TMP5
023510 010037 001642                            MOV     R0,$TMP6
023514 005037 001644                            CLR     $TMP7
023520 010037 001656                            MOV     R0,$TMP14
023524 062737 000002 001656                     ADD     #2,$TMP14
023532 005037 001660                            CLR     $TMP15
023536 011037 001646                            MOV     (R0),$TMP10
023542 016037 000002 001650                     MOV     2(R0),$TMP11
023550 010137 001652                            MOV     R1,$TMP12
023554 005037 001654                            CLR     $TMP13
023560 104045                           64$:    ERROR   45
023562 012737 177777 177744                     MOV     #-1,@#MEMERR

023570 010002                                   MOV     R0,R2
023572 020102                                   CMP     R1,R2
023574 001002                                   BNE     2$
023576 000137 023240                            JMP     EE4
023602 005202                           2$:     INC     R2
023604 020102                                   CMP     R1,R2
023606 001002                                   BNE     3$
023610 000137 023302                            JMP     EE7
023614 005202                           3$:     INC     R2
023616 020102                                   CMP     R1,R2
023620 001002                                   BNE     4$
023622 000137 023344                            JMP     EE10
023626 000137 023712                    4$:     JMP     EEDONE


```

```
023632  011637  001652                  EEERR3: MOV     (SP),$TMP12             ;REPORT FAILURE TO WRITE
                                                                                ;THE CORRECT BYTE
023636  010037  001632                          MOV     R0,$TMP2
023642  005037  001634                          CLR     $TMP3
023646  010037  001636                          MOV     R0,$TMP4
023652  062737  000002  001636                  ADD     #2,$TMP4
023660  005037  001640                          CLR     $TMP5
023664  011037  001642                          MOV     (R0),$TMP6
023670  016037  000002  001644                  MOV     2(R0),$TMP7
023676  010137  001646                          MOV     R1,$TMP10
023702  005037  001650                          CLR     $TMP11
023706  104047                                  ERROR   47
023710  000207                                  RTS     PC


023712  104416                          EEDONE: RSET                            ;DONE!

                                        ;;*****************************************************************
                                        ;*TEST 23       CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ZEROES TEST
                                        ;*
                                        ;*THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC.
                                        ;*HERE UNIBUS DATOB'S WILL MOVE ZEROES INTO A BACKROUND
                                        ;*PATTERN OF ONES.
                                        ;*
                                        ;;*****************************************************************
023714  000004                          TST23:  SCOPE
023716  012737  000010  001676                  MOV     #10,$TIMES              ;;DO 10 ITERATIONS
        000023                          HH=$TN-1
                                                                                ;SET THE SKAD REGISTER
023724  012737  024614  054230                  MOV     #TST24,SKAD             ;IN CASE THE TEST ABORTS.

023732  113737  001502  001626                  MOVB    $TSTNM,$TMP0

023740  104422                                  MMSKIP

023742  012737  024320  000114                  MOV     #HHERR1,@#CACHVEC


023750  012700  172340                          MOV     #KIPAR0,R0              ;SET UP MEMORY MANAGEMENT
                                                                                ;TO RELOCATE EVERYTHING
023754  012702  172300                          MOV     #KIPDR0,R2              ;THROUGH THE UNIBUS
023760  012703  000007                          MOV     #7,R3                   ;MAP PASSIVELY TO MEMORY.
023764  005004                                  CLR     R4                      ;BY PASSIVELY IS MEANT
023766  012705  170200                          MOV     #MAPL00,R5              ;THAT ADDRESS ARE
                                                                                ;RELOCATED TO THEMSELVES.
023772  012722  077406                  64$:    MOV     #77406,(R2)+
023776  010401                                  MOV     R4,R1
024000  072127  000006                          ASH     #6,R1
024004  010125                                  MOV     R1,(R5)+
024006  005025                                  CLR     (R5)+
024010  010410                                  MOV     R4,(R0)
024012  062720  170000                          ADD     #170000,(R0)+
024016  062704  000200                          ADD     #200,R4
024022  077315                                  SOB     R3,64$
024024  012710  177600                          MOV     #177600,(R0)
024030  012712  077406                          MOV     #77406,(R2)
```

```

024034  012737  000060  172516          MOV     #60,@#MMR3        ;TURN ON MEMORY MANAGEMENT
024042  012737  000001  177572          MOV     #1,@#MMR0         ;AND MAPPING BOX RELOCATION.

024050  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;FORCE MISSES.

024056  012700  024314                  MOV     #HHTMP2,R0        ;LOCATE THE TEST SPACE.
024062  042700  000003                  BIC     #3,R0
024066  010001                          MOV     R0,R1

024070  012710  177777          HH1:    MOV     #-1,(R0)
024074  012760  177777  000002          MOV     #-1,2(R0)
024102  000240                          NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024104  105011                          CLRB    (R1)
024106  022710  177400                  CMP     #177400,(R0)
024112  001403                          BEQ     HH3
024114  004737  024532          HH2:    JSR     PC,HHERR3
024120  000404                          BR      HH4
024122  022760  177777  000002  HH3:    CMP     #-1,2(R0)
024130  001371                          BNE     HH2

024132  005201                  HH4:    INC     R1
024134  012710  177777                  MOV     #-1,(R0)
024140  012760  177777  000002          MOV     #-1,2(R0)
024146  000240                          NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024150  105011                          CLRB    (R1)
024152  022710  000377                  CMP     #377,(R0)
024156  001403                          BEQ     HH6
024160  004737  024532          HH5:    JSR     PC,HHERR3
024164  000404                          BR      HH7
024166  022760  177777  000002  HH6:    CMP     #-1,2(R0)
024174  001371                          BNE     HH5

024176  005201                  HH7:    INC     R1
024200  012710  177777                  MOV     #-1,(R0)
024204  012760  177777  000002          MOV     #-1,2(R0)
024212  000240                          NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024214  105011                          CLRB    (R1)
024216  122760  177400  000002          CMPB    #177400,2(R0)
024224  001403                          BEQ     HH9
024226  004737  024532          HH8:    JSR     PC,HHERR3
024232  000403                          BR      HH10
024234  022710  177777          HH9:    CMP     #-1,(R0)
024240  001372                          BNE     HH8

024242  005201                  HH10:   INC     R1
024244  012710  177777                  MOV     #-1,(R0)
024250  012760  177777  000002          MOV     #-1,2(R0)
024256  000240                          NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024260  105011                          CLRB    (R1)
024262  022760  000377  000002          CMP     #377,2(R0)
024270  001403                          BEQ     HH12
024272  004737  024532          HH11:   JSR     PC,HHERR3
024276  000403                          BR      HH13
024300  022710  177777          HH12:   CMP     #-1,(R0)
024304  001372                          BNE     HH11
```

```

024306  000137  024612          HH13:     JMP     HHDONE

024312  000000                  HHTMP1:   .WORD   0
024314  000000                  HHTMP2:   .WORD   0                 ;THE TEST AREA
024316  000000                            .WORD   0


024320  032737  000002  177744  HHERR1:   BIT     #2,@#MEMERR       ;SHOULD BE A MAIN MEMORY
024326  001002                            BNE     1$                ;ADDRESS AND CONTROL LINE
024330  000137  054076                    JMP     SPUR              ;PARITY ERROR.
024334  020137  177740          1$:       CMP     R1,@#LOADRS       ;ERROR ADDRESS SHOULD BE
024340  001402                            BEQ     HHERR2            ;TEST ADDRESS.
024342  000137  054076                    JMP     SPUR
024346  012637  001642          HHERR2:   MOV     (SP)+,$TMP6
024352  005037  001664                    CLR     $TMP17
024356  005726                            TST     (SP)+             ;RESET THE STACK
024360  012737  000051  001666            MOV     #51,$TMP20
024366  013737  177740  001634            MOV     @#LOADRS,$TMP3
024374  013737  177742  001636            MOV     @#HIADRS,$TMP4
024402  013737  177744  001640            MOV     @#MEMERR,$TMP5
024410  010037  001642                    MOV     R0,$TMP6
024414  005037  001644                    CLR     $TMP7
024420  010037  001656                    MOV     R0,$TMP14
024424  062737  000002  001656            ADD     #2,$TMP14
024432  005037  001660                    CLR     $TMP15
024436  011037  001646                    MOV     (R0),$TMP10
024442  016037  000002  001650            MOV     2(R0),$TMP11
024450  010137  001652                    MOV     R1,$TMP12
024454  005037  001654                    CLR     $TMP13
024460  104051                  64$:      ERROR   51
024462  012737  177777  177744            MOV     #-1,@#MEMERR

024470  010002                            MOV     R0,R2
024472  020102                            CMP     R1,R2
024474  001002                            BNE     2$
024476  000137  024132                    JMP     HH4
024502  005202                  2$:       INC     R2
024504  020102                            CMP     R1,R2
024506  001002                            BNE     3$
024510  000137  024176                    JMP     HH7
024514  005202                  3$:       INC     R2
024516  020102                            CMP     R1,R2
024520  001002                            BNE     4$
024522  000137  024242                    JMP     HH10
024526  000137  024612          4$:       JMP     HHDONE


024532  011637  001652          HHERR3:   MOV     (SP),$TMP12       ;REPORT FAILURE TO WRITE
                                                                    ;THE CORRECT BYTE
024536  010037  001632                    MOV     R0,$TMP2
024542  005037  001634                    CLR     $TMP3
024546  010037  001636                    MOV     R0,$TMP4
024552  062737  000002  001636            ADD     #2,$TMP4
024560  005037  001640                    CLR     $TMP5
024564  011037  001642                    MOV     (R0),$TMP6
```

```
024570  016037  000002  001644          MOV     2(R0),$TMP7
024576  010137  001646                  MOV     R1,$TMP10
024602  005037  001650                  CLR     $TMP11
024606  104053                          ERROR   53
024610  000207                          RTS     PC


024612  104416                  HHDONE: RSET                    ;DONE!


                                ;;*********************************************************************
                                ;*TEST 24        CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST
                                ;*
                                ;*THIS TEST IS EXECUTED OPTIONALLY, ON THE CONDITION THAT
                                ;*BIT 12 OF THE SWITCH REGISTER IS ON WHEN PROGRAM CONTROL
                                ;*REACHES THIS POINT.  IF THIS SWITCH IS OFF THEN CONTROL
                                ;*IS PASSED TO THE NEXT TEST.  THIS IS DONE BECAUSE THIS
                                ;*TEST REQUIRES OPERATOR INTERVENTION.  THE USER IS ASKED TO
                                ;*GO THROUGH A POWER DOWN-POWER UP SEQUENCE.  THEN
                                ;*A SIMPLE SCAN IS MADE OF MEMORY WHICH CAUSES ALL
                                ;*DATA AND ADDRESS MEMORY LOCATIONS IN THE CACHE TO BE
                                ;*PARITY CHECKED.  IF THE POWER UP-CACHE INVLIDATER LOGIC
                                ;*WORKED NO PARITY ERRORS CAN OCCUR.  BUT IF THIS INVALIDATER
                                ;*FAILED THERE IS AN EXTREMELY HIGH PROBABILITY FOR THE
                                ;*OCCURENCE OF A CACHE DATA OR CACHE ADDRESS PARITY ERROR.
                                ;*IN FACT IF THE INVALIDATER CIRCUIT IS COMPLETELY INOPERATIVE
                                ;*IT WILL BE VIRTUALLY IMPOSSIBLE TO RESTART THE PROGRAM.
                                ;*WHEREAS MINOR OR NO FAILURES CAN AND WILL BE REPORTED.
                                ;*IF NO PARITY ERRORS ARE ENCOUNTERED THE USER WILL
                                ;*BE NOTIFIED SO THAT HE CAN KNOW IF A FATAL FAILURE
                                ;*HAS OCCURRED.
                                ;*
                                ;;*********************************************************************
024614  000004                  TST24:  SCOPE
        000024                  DD=$TN-1
                                                                ;SET THE SKAD REGISTER
024616  012737  025050  054230          MOV     #TST25,SKAD     ;IN CASE THE TEST ABORTS.

024624  113737  001502  001626          MOVB    $TSTNM,$TMP0
024632  012737  054076  000114          MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.

024640  032737  010000  177570          BIT     #SW12,@#SWR     ;SEE IF THE USER HAS CHOSEN
024646  001002                          BNE     DD1             ;TO RUN THIS TEST, SW12=1.
024650  000177  027354                  JMP     @SKAD           ;NO, SO GO TO NEXT TEST.

024654  012737  025006  000114  DD1:    MOV     #DDPER,@#CACHVEC        ;YES, SO SET UP THE PARITY
                                                                ;ERROR VECTOR.
024662  013737  000024  025036          MOV     @#24,DDTMP      ;SAVE THE OLD CONTENTS
024670  012737  024710  000024          MOV     #DDPD,@#24      ;OF THE PWER FAIL TRAP
024676  005037  025040                  CLR     DDCNTR          ;VECTOR AND RESET THIS
                                                                ;VECTOR. CLEAR AN ERROR COUNT.
024702  104400                          TYPE                    ;TELL THE USER TO POWER
024704  065536                          .WORD   PDMSG1          ;DOWN.
024706  000777                          BR      .               ;WAIT, SHOULD THIS
                                                                ;WAIT TIME OUT????
024710  000240                  DDPD:   NOP                     ;FOR SCOPE SYNC!
```

```
024712  012737  024722  000024            MOV     #DDPV,@#24          ;POWER DOWN ROUTINE
024720  000777                            BR      .                   ;JUST SET UP FOR POWER UP.
024722  012706  001500          DDPV:     MOV     #STACK,SP           ;RESET THE STACK POINTER
024726  013737  025036  000024            MOV     DDTMP,@#24          ;RESET POWER FAIL VECTOR.
024734  005000                            CLR     R0                  ;SET UP FOR SCAN.
024736  012701  001000                    MOV     #1000,R1
024742  005720                  1$:       TST     (R0)+
024744  077102                            SOB     R1,1$
024746  013737  025036  000024  DDPU1:    MOV     DDTMP,@#24          ;RESET THE POWER FAIL VECTOR.
024754  005737  025040                    TST     DDCNTR              ;WERE THERE ANY ERRORS?
024760  001004                            BNE     DDPU2
024762  104400                            TYPE                        ;NO
024764  065714                            .WORD   PDMSG2
024766  000137  025042                    JMP     DDDONE

024772                          DDPU2:                                ;REPORT ERROR SUMMARY
024772  013737  025040  001632            MOV     DDCNTR,$TMP2
025000  104054                  1$:       ERROR   54
025002  000137  025042                    JMP     DDDONE

025006  032737  000360  177744  DDPER:    BIT     #360,@#MEMERR       ;THE ERROR SHOULD BE
025014  001406                            BEQ     DDPER1              ;A CACHE ADDRESS OR CACHE
025016  012737  177777  177744            MOV     #-1,MEMERR          ;DATA PARITY ERROR
025024  005237  025040                    INC     DDCNTR
025030  000002                            RTI

025032  000137  054076          DDPER1: JMP       SPUR

025036  000000                  DDTMP:    .WORD   0                   ;STORAGE FOR POWER FAIL
                                                                      ;VECTORS OLD PC
025040  000000                  DDCNTR: .WORD     0                   ;ERROR COUNT.

025042  104416                  DDDONE: RSET
025044  012706  001500                    MOV     #STACK,SP

                                ;;*****************************************************************
                                ;;*TEST 25          CACHE DATA MULTIPLEXER, CDMX, TEST
                                ;;*
                                ;;*THIS TEST PUTS DIFFERENT PATTERNS OF DATA AT THE INPUTS
                                ;;*OF THE CDMX AND TESTS FOR PROPER SELECTION AND GOOD DATA.
                                ;;*
                                ;;*****************************************************************
025050  000004                  TST25:    SCOPE
025052  012737  000010  001676            MOV     #10,$TIMES          ;;DO 10 ITERATIONS
                                                                      ;SET THE SKAD REGISTER
025060  012737  026156  054230            MOV     #TST26,SKAD         ;IN CASE THE TEST ABORTS.

025066  012737  054076  000114            MOV     #SPUR,@#CACHVEC ;PREPARE FOR UNEXPECTED ERRORS.
025074  113737  001502  001626            MOVB    $TSTNM,$TMP0
025102  012705  000006                    MOV     #6,R5               ;INITIALIZE
025106  012737  000004  026130            MOV     #4,JJCNT
025114  012700  026146                    MOV     #JJTMP2,R0
025120  042700  176002                    BIC     #176002,R0
025124  012701  140000                    MOV     #TESTR1,R1
025130  060001                            ADD     R0,R1
025132  012702  142000                    MOV     #TESTR2,R2
```

```
025136  060002                                ADD     R0,R2
025140  012703  144000                        MOV     #TESTR3,R3
025144  060003                                ADD     R0,R3
025146  012704  026134                        MOV     #JJPAT2,R4

025152  012737  125252  026132                MOV     #125252,JJPAT1  ;JJPAT1 CONTAINS THE DATA
                                                                      ;WHICH WILL ENTER THE
                                                                      ;MAIN MEMORY EVEN INPUTS
                                                                      ;TO THE CDMX. INITIALLY
                                                                      ;THIS WILL BE 125252
025160  012737  052525  026134                MOV     #52525,JJPAT2   ;DATA FOR MAIN MEMORY ODD
                                                                      ;WORD INPUT TO CDMX
025166  005037  026136                        CLR     JJPAT3          ;GROUP 0 DATA INPUTS TO CDMX.
025172  012737  177777  026140                MOV     #-1,JJPAT4      ;GROUP 1 DATA INPUTS TO CDMX.
025200  012737  025200  001512  JJ1:          MOV     #JJ1,$LPERR
025206  013713  026132                        MOV     JJPAT1,(R3)     ;WRITE THE MAIN MEMORY
025212  013763  026134  000002                MOV     JJPAT2,2(R3)    ;EVEN AND ODD WORD PATTERNS

025220  012737  000034  177746                MOV     #S0M0M1,@#CONTRL        ;WRITE THE GROUP ZERO
025226  013711  026136                        MOV     JJPAT3,(R1)     ;PATTERN
025232  013761  026136  177776                MOV     JJPAT3,-2(R1)
025240  013761  026136  000002                MOV     JJPAT3,2(R1)
025246  005711                                TST     (R1)
025250  012737  000054  177746                MOV     #S1M0M1,@#CONTRL        ;WRITE THE GROUP ONE PATTERN
025256  013712  026140                        MOV     JJPAT4,(R2)
025262  013762  026140  177776                MOV     JJPAT4,-2(R2)
025270  013762  026140  000002                MOV     JJPAT4,2(R2)
025276  005712                                TST     (R2)

025300  005037  177746                        CLR     @#CONTRL
025304  000240                                NOP
025306                          JJ2:
025306  000240                                NOP
025310  016100  000000                        MOV     0(R1),R0
025314  032737  000010  177752                BIT     #10,@#HITMIS    ;MUST BE A HIT!
025322  001011                                BNE     JJ3
025324  012737  000000  001630                MOV     #0,$TMP1
025332  010137  001632                        MOV     R1,$TMP2
025336  062737  000000  001632                ADD     #0,$TMP2
025344  104001                  66$:          ERROR   1
025346  020037  026136          JJ3:          CMP     R0,JJPAT3
025352  001406                                BEQ     65$
025354  012737  025366  001630                MOV     #64$,$TMP1
025362  010037  001632                        MOV     R0,$TMP2
025366  104005                  64$:          ERROR   5
025370                          65$:
025370  012737  025376  001512                MOV     #JJ4,$LPERR
025376                          JJ4:
025376  000240                                NOP
025400  016100  000002                        MOV     2(R1),R0
025404  032737  000010  177752                BIT     #10,@#HITMIS    ;MUST BE A HIT!
025412  001011                                BNE     JJ5
025414  012737  000000  001630                MOV     #0,$TMP1
025422  010137  001632                        MOV     R1,$TMP2
025426  062737  000002  001632                ADD     #2,$TMP2
025434  104001                  66$:          ERROR   1
```

```
025436  020037  026136          JJ5:    CMP     R0,JJPAT3
025442  001406                          BEQ     65$
025444  012737  025456  001630          MOV     #64$,$TMP1
025452  010037  001632                  MOV     R0,$TMP2
025456  104005                  64$:    ERROR   5
025460                          65$:
025460  012737  025466  001512          MOV     #JJ6,$LPERR
025466                          JJ6:
025466  000240                          NOP
025470  016200  000000                  MOV     0(R2),R0
025474  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
025502  001011                          BNE     JJ7
025504  012737  000001  001630          MOV     #1,$TMP1
025512  010237  001632                  MOV     R2,$TMP2
025516  062737  000000  001632          ADD     #0,$TMP2
025524  104001                  66$:    ERROR   1
025526  020037  026140          JJ7:    CMP     R0,JJPAT4
025532  001406                          BEQ     65$
025534  012737  025546  001630          MOV     #64$,$TMP1
025542  010037  001632                  MOV     R0,$TMP2
025546  104006                  64$:    ERROR   6
025550                          65$:
025550  012737  025556  001512          MOV     #JJ8,$LPERR
025556                          JJ8:
025556  000240                          NOP
025560  016200  000002                  MOV     2(R2),R0
025564  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
025572  001011                          BNE     JJ9
025574  012737  000001  001630          MOV     #1,$TMP1
025602  010237  001632                  MOV     R2,$TMP2
025606  062737  000002  001632          ADD     #2,$TMP2
025614  104001                  66$:    ERROR   1
025616  020037  026140          JJ9:    CMP     R0,JJPAT4
025622  001406                          BEQ     65$
025624  012737  025636  001630          MOV     #64$,$TMP1
025632  010037  001632                  MOV     R0,$TMP2
025636  104006                  64$:    ERROR   6
025640                          65$:
025640  012737  025646  001512          MOV     #JJ10,$LPERR
025646  000240                  JJ10:   NOP
025650  012737  000014  177746          MOV     #M1M0,@#CONTRL  ;CHECK MAIN MEMORY DATA
025656  011300                          MOV     (R3),R0         ;EVEN WORD
025660  020037  026132                  CMP     R0,JJPAT1
025664  001403                          BEQ     1$
025666  010037  001632                  MOV     R0,$TMP2
025672  104007                          ERROR   7
025674  012737  025702  001512  1$:     MOV     #JJ11,$LPERR
025702  016300  000002          JJ11:   MOV     2(R3),R0        ;CHECK MAIN MEMORY EVEN
025706  020037  026134                  CMP     R0,JJPAT2       ;WORD
025712  001403                          BEQ     JJ12
025714  010037  001632                  MOV     R0,$TMP2
025720  104010                  1$:     ERROR   10

025722  005037  177746          JJ12:   CLR     @#CONTRL
025726  020427  026140                  CMP     R4,#JJPAT4      ;NOW GET EVERY PERMUTATION
025732  001011                          BNE     JJ13            ;OF THE FOUR TEST PATTERNS:
```

```
                                                                  ;125252,052525,177777 AND
025734  011437  026142                   MOV     (R4),JJPAT5     ;000000 INTO JJPAT1, JJPAT2,
025740  013714  026134                   MOV     JJPAT2,(R4)     ;JJPAT3 AND JJPAT4 AND
025744  012704  026134                   MOV     #JJPAT2,R4      ;REPEAT THE TEST.
025750  013714  026142                   MOV     JJPAT5,(R4)
025754  000406                           BR      JJ14

025756  012437  026142           JJ13:   MOV     (R4)+,JJPAT5
025762  011464  177776                   MOV     (R4),-2(R4)
025766  013714  026142                   MOV     JJPAT5,(R4)

025772  005305                   JJ14:   DEC     R5
025774  001402                           BEQ     1$
025776  000137  025200                   JMP     JJ1
026002  012705  000006           1$:     MOV     #6,R5
026006  013737  026132  026142           MOV     JJPAT1,JJPAT5
026014  005337  026130                   DEC     JJCNT

026020  023727  026130  000003           CMP     JJCNT,#3
026026  001010                           BNE     JJ15
026030  013737  026134  026132           MOV     JJPAT2,JJPAT1
026036  013737  026142  026134           MOV     JJPAT5,JJPAT2
026044  000137  025200                   JMP     JJ1

026050  023727  026130  000002   JJ15:   CMP     JJCNT,#2
026056  001010                           BNE     JJ16
026060  013737  026136  026132           MOV     JJPAT3,JJPAT1
026066  013737  026142  026136           MOV     JJPAT5,JJPAT3
026074  000137  025200                   JMP     JJ1

026100  023727  026130  000001   JJ16:   CMP     JJCNT,#1
026106  001023                           BNE     JJ17            ;DONE?
026110  013737  026140  026132           MOV     JJPAT4,JJPAT1
026116  013737  026142  026140           MOV     JJPAT5,JJPAT4
026124  000137  025200                   JMP     JJ1

026130  000000                   JJCNT:  .WORD   0               ;COUNTER USED TO GENERATE
                                                                  ;PERMUTATIONS.
026132  000000                   JJPAT1: .WORD   0               ;MAIN MEMORY EVEN WORD DATA PATTERN
026134  000000                   JJPAT2: .WORD   0               ;MAIN MEMORY ODD WORD DATA PATTERN
026136  000000                   JJPAT3: .WORD   0               ;GROUP 0 DATA PATTERN
026140  000000                   JJPAT4: .WORD   0               ;GROUP 1 DATA PATTERN
026142  000000                   JJPAT5: .WORD   0               ;TEMPORARY STORAGE

026144  000000                   JJTMP1: .WORD   0               ;TEST AREA, SO CODE WON'T
026146  000000  000000  000000   JJTMP2: .WORD   0,0,0,0         ;OVER LAP THE HITS OF
026154  000000
                                                                  ;THE TEST WORDS.

026156                           JJ17:                           ;DONE!

                                 ;;*************************************************************
                                 ;*TEST 26       CACHE DATA MEMORY ADDRESS DRIVERS TEST
                                 ;*
                                 ;*THIS TEST PERFORMS A DUAL ADDRESS TEST ON THE
                                 ;*CACHE DATA MEMORIES OF BOTH GROUPS.
```

```
                                        ;*
                                        ;;**********************************************************************
026156  000004                          TST26:  SCOPE
026160  012737  000010  001676                  MOV     #10,$TIMES          ;;DO 10 ITERATIONS
                                                                            ;SET THE SKAD REGISTER
026166  012737  026666  054230                  MOV     #TST27,SKAD         ;IN CASE THE TEST ABORTS.

026174  012737  054076  000114                  MOV     #SPUR,@#CACHVEC
026202  113737  001502  001626                  MOVB    $TSTNM,$TMP0

026210  012737  000001  026660          GG1:    MOV     #1,GGFLG1           ;INITIALIZE FOR A TEST
026216  012737  000054  026662                  MOV     #S1M0M1,GGGS        ;ON GROUP 1 FIRST
026224  012737  000034  026664                  MOV     #S0M0M1,GGGM        ;S0M1 AND S1M0 ARE PATTERNS
                                                                            ;DESTINED FOR THE CACHE
                                                                            ;CONTROL REGISTER
026232  012700  026232                  GG2:    MOV     #GG2,R0             ;MAKE THIS CODE, LOCATIONS
026236  012701  001000                          MOV     #1000,R1            ;GG2 THROUGH GG2+2000(OCT),
026242  013737  026662  177746          GG3:    MOV     GGGS,@#CONTRL       ;HITS IN THE GROUP NOT
026250  005760  002000                          TST     2000(R0)            ;BEING TESTED AND MISSES
026254  013737  026664  177746                  MOV     GGGM,@#CONTRL       ;IN THE GROUP BEING TESTED.
026262  005720                                  TST     (R0)+
026264  077112                                  SOB     R1,GG3
026266  013700  026662                          MOV     GGGS,R0             ;MAKE THE TEST AREA
026272  042700  177717                          BIC     #177717,R0          ;HITS IN THE GROUP
026276  010037  177746                          MOV     R0,@#CONTRL         ;BEING TESTED
026302  012701  140000                          MOV     #TESTR1,R1
026306  012700  001000                          MOV     #1000,R0
026312  012737  026320  001512                  MOV     #GG4,$LPERR
026320  000240                          GG4:    NOP
026322  005011                                  CLR     (R1)
026324  005711                                  TST     (R1)
026326  005711                                  TST     (R1)
026330  032737  000010  177752                  BIT     #10,@#HITMISS
026336  001006                                  BNE     2$
026340  013737  026660  001630                  MOV     GGFLG1,$TMP1
026346  010137  001632                          MOV     R1,$TMP2
026352  104001                          1$:     ERROR   1
026354  005721                          2$:     TST     (R1)+
026356  077020                                  SOB     R0,GG4
026360  013700  026664                          MOV     GGGM,R0             ;FROM HERE ON SELECT
026364  042700  177717                          BIC     #177717,R0          ;THE GROUP NOT BEING
026370  010037  177746                          MOV     R0,@#CONTRL         ;TESTED

026374  012701  140000                          MOV     #TESTR1,R1
026400  012700  001000                          MOV     #1000,R0
026404  012737  026412  001512                  MOV     #GG5,$LPERR
026412  000240                          GG5:    NOP                         ;
026414  010111                                  MOV     R1,(R1)             ;WRITE #ADDRESS INTO @#ADDRESS.
026416  005721                                  TST     (R1)+
026420  077004                                  SOB     R0,GG5

026422  012701  140000                          MOV     #TESTR1,R1
026426  012700  001000                          MOV     #1000,R0
026432  012737  026440  001512                  MOV     #GG6,$LPERR
026440  000240                          GG6:    NOP
026442  011102                                  MOV     (R1),R2             ;READ BACK THE ADDRESS
```

```
 5075  026444  032737  000010  177752          BIT     #10,@#HITMIS
 5076  026452  001006                          BNE     GG7
 5077  026454  013737  026660  001630          MOV     GGFLG1,$TMP1
```

```
026462  010137  001632                    MOV     R1,$TMP2
026466  104001                    1$:     ERROR   1

026470  020102                    GG7:    CMP     R1,R2           ;DOES @#ADDRESS CONTAIN
026472  001412                            BEQ     GG8             ;#ADDRESS

026474  013737  026660  001630            MOV     GGFLG1,$TMP1
026502  010137  001632                    MOV     R1,$TMP2
026506  010237  001634                    MOV     R2,$TMP3
026512  010137  001636                    MOV     R1,$TMP4
026516  104016                    1$:     ERROR   16

026520  005121                    GG8:    COM     (R1)+           ;COMPLIMENT DATA
026522  077032                            SOB     R0,GG6          ;LOOP FOR NEXT ADDRESS.
026524  012701  140000                    MOV     #TESTR1,R1
026530  012700  001000                    MOV     #1000,R0
026534  012737  026542  001512            MOV     #GG9,$LPERR
026542  000240                    GG9:    NOP
026544  011102                            MOV     (R1),R2         ;GO BACK AND CHECK
026546  032737  000010  177752            BIT     #10,@#HITMIS    ;COMPLIMENTED DATA
026554  001006                            BNE     GG10
026556  013737  026660  001630            MOV     GGFLG1,$TMP1
026564  010137  001632                    MOV     R1,$TMP2
026570  104001                    1$:     ERROR   1
                                                                  ;?????

026572  010103                    GG10:   MOV     R1,R3           ;IS COMPLIMENT DATA CORRECT?
026574  005103                            COM     R3
026576  020302                            CMP     R3,R2
026600  001412                            BEQ     GG11
026602  013737  026660  001630            MOV     GGFLG1,$TMP1
026610  010337  001632                    MOV     R3,$TMP2
026614  010237  001634                    MOV     R2,$TMP3
026620  010137  001636                    MOV     R1,$TMP4
026624  104016                    1$:     ERROR   16

026626  005721                    GG11:   TST     (R1)+           ;TEST NEXT LOCATION
026630  077034                            SOB     R0,GG9

026632  012737  000034  026662            MOV     #S0M0M1,GGGS    ;GO BACK AND RUN
026640  012737  000054  026664            MOV     #S1M0M1,GGGM    ;TEST IN GROUP 0.
026646  005337  026660                    DEC     GGFLG1
026652  001005                            BNE     GG12
026654  000137  026232                    JMP     GG2

026660  000000                    GGFLG1: .WORD   0               ;GROUP BEING TESTED, 0 OR 1.

026662  000000                    GGGS:   .WORD   0               ;CACHE CONTROL REGISTER
026664  000000                    GGGM:   .WORD   0               ;PATTERNS

026666                            GG12:                           ;DONE!

                                  ;;*****************************************************************
                                  ;*TEST 27        CACHE DATA MEMORY COUNT PATTERN TEST
                                  ;*
                                  ;*THIS TEST RUNS A COUNT PATTERN THROUGH EACH LOCATION
```

```
                                        ;*OF THE CACHE DATA MEMORY FOR EACH GROUP.
                                        ;*
                                        ;;*******************************************************************
026666  000004                          TST27:  SCOPE
026670  012737  000010  001676                  MOV     #10,$TIMES          ;;DO 10 ITERATIONS
                                                                            ;SET THE SKAD REGISTER
026676  012737  027646  054230                  MOV     #TST30,SKAD         ;IN CASE THE TEST ABORTS.

026704  012737  054076  000114                  MOV     #SPUR,@#CACHVEC
026712  113737  001502  001626                  MOVB    $TSTNM,$TMP0

026720  012737  000001  027354          LL1:    MOV     #1,LLFLG1           ;TEST GROUP ONE FIRST
026726  012737  000044  027362                  MOV     #S1M0,LLGS          ;S1M0 AND S0M1 ARE PATTERNS
026734  012737  000030  027364                  MOV     #S0M1,LLGM          ;WHICH WILL BE LOADED INTO
026742  012737  026742  001512          LL2:    MOV     #LL2,$LPERR         ;THE CACHE CONTROL REGISTER.
026750  012737  054076  000114                  MOV     #SPUR,@#CACHVEC
026756  012700  026742                          MOV     #LL2,R0             ;MAKE THIS CODE, LOCATIONS
026762  012701  001000                          MOV     #1000,R1            ;LL2 THROUGH LL2+2000 (OCT)
                                                                            ;HITS IN THE CACHE GROUP
026766  013737  027364  177746          LL3:    MOV     LLGM,@#CONTRL       ;NOT BEING TESTED, AND MISSES
026774  005710                                  TST     (R0)                ;TO THE CACHE GROUP BEING
026776  013737  027362  177746                  MOV     LLGS,@#CONTRL       ;TESTED.
027004  005760  002000                          TST     2000(R0)
027010  062700  000002                          ADD     #2,R0
027014  077114                                  SOB     R1,LL3

027016  012701  140000                          MOV     #TESTR1,R1          ;MAKE THE MEMORY TEST AREA
027022  012700  001000                          MOV     #1000,R0            ;HITS IN THE GROUP BEING
027026  012737  027050  001512                  MOV     #1$,$LPERR          ;TESTED.
027034  013702  027362                          MOV     LLGS,R2
027040  042702  177717                          BIC     #177717,R2
027044  010237  177746                          MOV     R2,@#CONTRL
027050  005011                          1$:     CLR     (R1)
027052  005711                                  TST     (R1)
027054  005721                                  TST     (R1)+
027056  032737  000010  177752                  BIT     #10,@#HITMIS
027064  001011                                  BNE     3$
027066  013737  027354  001630                  MOV     LLFLG1,$TMP1
027074  011137  001632                          MOV     (R1),$TMP2
027100  062737  177776  001632                  ADD     #-2,$TMP2
027106  104001                          2$:     ERROR   1
027110  077021                          3$:     SOB     R0,1$
027112  013700  027364                          MOV     LLGM,R0             ;FROM NOW ON SELECT
027116  042700  177717                          BIC     #177717,R0          ;THE GROUP NOT BEING
027122  010037  177746                          MOV     R0,@#CONTRL         ;TESTED

027126  012701  140000                          MOV     #TESTR1,R1          ;INITIALIZE FOR TEST.
027132  012700  001000                          MOV     #1000,R0            ;COUNTER.
027136  005002                          LL4:    CLR     R2                  ;DATA PATTERN WRITTEN
027140  005003                                  CLR     R3                  ;LOGICAL 'OR' OF BAD DATA
027142  012704  177777                          MOV     #177777,R4          ;LOGICAL 'AND' OF BAD DATA
027146  005005                                  CLR     R5                  ;DATA PATTERN READ
027150  005037  027366                          CLR     LLCNT1              ;NUMBER OF LOCATIONS WHICH FAIL.
027154  005037  027356                          CLR     LLFLG2              ;ERROR IN GROUP FLAG
027160  012737  027166  001512                  MOV     #LL5,$LPERR
027166  005037  027360                  LL5:    CLR     LLFLG4              ;ERROR IN TESTED WORD FLAG.
```

```
 027172  000240                           NOP                           ;FOR SCOPING WITH AN OSCILLOSCOPE.
 027174  010211                           MOV     R2,(R1)
 027176  011105                           MOV     (R1),R5
 027200  032737  000010  177752           BIT     #10,@#HITMIS
 027206  001006                           BNE     LL6
 027210  013737  027354  001630           MOV     LLFLG1,$TMP1
 027216  010137  001632                   MOV     R1,$TMP2
 027222  104001                   1$:     ERROR   1
 027224  020205                   LL6:    CMP     R2,R5                 ;GOOD DATA
 027226  001402                           BEQ     LL7
 027230  000137  027600                   JMP     LLERR2                ;BAD DATA BUT NO TRAP OR
                                                                        ;ABORT OCCURRED!
 027234                           LL7:                                  ;DECREMENT THE COUNT PATTERN
                                                                        ;AND LOOP IF NOT DONE
 027234  005737  027360                   TST     LLFLG4                ;IF THERE WAS AN ERROR
 027240  001405                           BEQ     LL8                   ;IN THE WORD JUST TESTED
 027242  005237  027366                   INC     LLCNT1                ;INCREMENT LLCNT1
 027246  012737  177777  027356           MOV     #-1,LLFLG2            ;AND SET ERROR IN GROUP FLAG.
 027254  062701  000002           LL8:    ADD     #2,R1                 ;GO TO NEXT WORD.
 027260  077036                           SOB     R0,LL5

 027262  005737  027356                   TST     LLFLG2                ;DONE WITH THAT GROUP,
 027266  001417                           BEQ     LL9                   ;SEE IF THERE WERE
 027270  112737  000013  001516           MOVB    #13,$ITEMB            ;ANY ERRORS. IF SO THEN
 027276  013737  027354  001630           MOV     LLFLG1,$TMP1          ;PRINT AN ERROR SUMMARY
 027304  010437  001632                   MOV     R4,$TMP2              ;FOR THAT GROUP.
 027310  010337  001634                   MOV     R3,$TMP3
 027314  013737  027366  001636           MOV     LLCNT1,$TMP4
 027322  004737  054744                   JSR     PC,ERTYPE

 027326  012737  000044  027364   LL9:    MOV     #S1M0,LLGM            ;TEST THE OTHER GROUP, 0.
 027334  012737  000030  027362           MOV     #S0M1,LLGS            ;
 027342  005337  027354                   DEC     LLFLG1
 027346  001137                           BNE     LL10                  ;DONE?
 027350  000137  026742                   JMP     LL2

 027354  000000                   LLFLG1: .WORD   0                     ;GROUP BEING TESTED, 1 OR 0.
 027356  000000                   LLFLG2: .WORD   0                     ;ERROR OCCURRED IN GROUP FLAG.

 027360  000000                   LLFLG4: .WORD   0                     ;ERROR OCCURRED IN WORD FLAG.

 027362  000000                   LLGS:   .WORD   0                     ;PATTERNS FOR CONTROL REGISTER
 027364  000000                   LLGM:   .WORD   0

 027366  000000                   LLCNT1: .WORD   0                     ;GROUP ERROR COUNT

 027370  000000                   LLMER:  .WORD   0                     ;TEMPORARY STORAGE FOR
                                                                        ;THE CACHE ERROR REGISTER.
 027372  000000                   LLTMP1: .WORD   0

 027374  013737  177744  027370   LLERR1: MOV     @#MEMERR,LLMER        ;COME HERE ON PARITY
 027402  012737  004100  027372           MOV     #4100,LLTMP1          ;ABORT OR TRAP.
 027410  005737  027354                   TST     LLFLG1                ;TESTING GROUP 1 OR 0?
 027414  001403                           BEQ     1$
 027416  012737  004200  027372           MOV     #4200,LLTMP1
 027424  023737  027372  027370   1$:     CMP     LLTMP1,LLMER          ;WAS THE ERROR EXPECTED?
```

```
027432  001402                          BEQ     2$
027434  000137  054076                  JMP     SPUR                    ;NO!

027440  020137  177740          2$:     CMP     R1,@#LOADRS             ;WAS THAT ADDRESS EXPECTED?
027444  001402                          BEQ     3$
027446  000137  054076                  JMP     SPUR                    ;NO!

027452  012737  177777  027360  3$:     MOV     #-1,LLFLG4              ;SET WORD ERROR FLAG
027460  050203                          BIS     R2,R3                   ;DO 'OR' OF FAILIING DATA
027462  005102                          COM     R2
027464  040204                          BIC     R2,R4                   ;DO 'AND' OF FAILING DATA
027466  005102                          COM     R2
027470  011637  001630                  MOV     (SP),$TMP1
027474  022626                          CMP     (SP)+,(SP)+
027476  013737  027354  001632          MOV     LLFLG1,$TMP2
027504  010237  001634                  MOV     R2,$TMP3
027510  010137  001644                  MOV     R1,$TMP7
027514  013737  177740  001636          MOV     @#LOADRS,$TMP4
027522  013737  177742  001640          MOV     @#HIADRS,$TMP5
027530  042737  140000  001640          BIC     #140000,$TMP5
027536  013737  027370  001642          MOV     LLMER,$TMP6
027544  104011                          ERROR   11                      ;REPORT ERROR.

027546  012737  027560  000114          MOV     #LLERR3,@#CACHVEC               ;BEFORE CONTINUING THE
                                                                        ;BAD PARITY IN THE WORD
                                                                        ;BEING TESTED MUST BE
                                                                        ;DEALT WITH!
027554  005011                          CLR     (R1)                    ;THIS INSTRUCTION CLR (R1)
027556  005711                          TST     (R1)                    ;SHOULD TRAP!

027560  012737  177777  177744  LLERR3: MOV     #-1,@#MEMERR            ;CLR THE ERROR REGISTER
027566  012737  027374  000114          MOV     #LLERR1,@#CACHVEC               ;RESTORE THE PARITY ERROR
027574  000137  027234                  JMP     LL7                     ;VECTOR AND CONTINUE.

027600  012737  177777  027360  LLERR2: MOV     #-1,LLFLG4              ;BAD DATA WAS READ BUT
                                                                        ;NO TRAP OR ABORT OCCURRED!
027606  050203                          BIS     R2,R3                   ;'OR' BAD DATA
027610  005102                          COM     R2
027612  040204                          BIC     R2,R4                   ;'AND' BAD DATA
027614  005102                          COM     R2
027616  013737  027354  001630          MOV     LLFLG1,$TMP1
027624  010137  001634                  MOV     R1,$TMP3
027630  010237  001636                  MOV     R2,$TMP4
027634  010537  001640                  MOV     R5,$TMP5

027640  104012                  1$:     ERROR   12                      ;REPORT ERROR.

027642  000137  027234                  JMP     LL7                     ;CONTINUE TEST.
027646                          LL10:


                                ;;*****************************************************************
                                ;*TEST 30        CACHE DATA MEMORY PARITY CHECKERS LOW BYTE TEST
                                ;*
                                ;*THIS IS A TEST OF THE TWO CACHE DATA MEMORY PARITY
                                ;*CHECKERS FOR THE LOW BYTE, ONE FOR EACH GROUP. THE
```

```
                                          ;*MAINTENANCE REGISTER ISUSED TO FORCE A PARITY A
                                          ;*PARITY ERROR AT EVERY DATA PATTERN WHICH HAS A ONE
                                          ;*PARITY BIT. NOTE THAT THE CACHE DATA MEMORY PARITY HAS,
                                          ;*EFFECTIVELY, ODD PARITY. THE MAINTENANCE FUNCTION ON THE
                                          ;*CACHE DATA MEMORY PARITY CHECKERS HAS THE EFFECT OF
                                          ;*FORCING THE PARITY BIT OF THE BYTE BEING CHECKED TO
                                          ;*ZERO. THIS MEANS THAT ONCE THIS MAINTENANCE FUNCTION
                                          ;*IS ENABLED THE ERROR WILL OCCUR ON A SUBSEQUENT
                                          ;*READ OF A BYTE WITH A ONE PARITY BIT, THAT IS
                                          ;*BYTES WITH ZERO PARITY BITS WILL NOT CAUSE THE ERROR.
                                          ;*
                                          ;;*************************************************************************
027646  000004                            TST30:  SCOPE
027650  012737  000020  001676                    MOV     #20,$TIMES          ;;DO 20 ITERATIONS
        000031                            IIA=$TN
                                                                              ;SET THE SKAD REGISTER
027656  012737  030320  054230                    MOV     #TST31,SKAD         ;IN CASE THE TEST ABORTS.

027664  113737  001502  001626                    MOVB    $TSTNM,$TMP0
027672  012737  054076  000114                    MOV     #SPUR,@#CACHVEC

027700  005000                                    CLR     R0                  ;THIS IS THE COUNTER CONTAINING
                                                                              ;THE TEST DATA PATTERN
027702  012737  027702  001512            IIA1:   MOV     #IIA1,$LPERR
027710  004737  054470                            JSR     PC,PARCNT           ;SET IF THIS TEST PATTERN HAS
027714  032702  000001                            BIT     #BIT0,R2            ;THE PARITY BIT SET (1), IF NOT
027720  001402                                    BEQ     IIA2                ;GO TO THE NEXT PATTERN
027722  000137  030300                            JMP     IIA7
027726  012737  000030  177746            IIA2:   MOV     #SOM1,@#CONTRL      ;SELECT GROUP ZERO.
027734  012737  030204  000114                    MOV     #IIAR1,@#CACHVEC            ;SET UP FOR THE ERROR
027742  012705  030202                            MOV     #IIAT1,R5           ;MAKE THE TEST ADDRESS A
027746  005715                                    TST     (R5)                ;HIT IN GROUP ZERO
027750  005715                                    TST     (R5)                ;MAKE SURE IT IS A HIT

                                                                              ;SEE IF REFERENCE ADDRESS
027752  032737  000010  177752                    BIT     #10,@#HITMIS        ;IS A HIT.
027760  001007                                    BNE     1$
                                                                              ;IF NOT ERROR!
027762  010537  001632                            MOV     R5,$TMP2
027766  012737  000000  001630                    MOV     #0,$TMP1
027774  104001                                    ERROR   1

027776  104420                                    SKIPT                       ;ERROR FATAL. GO TO NEXT TEST.


030000  012704  000020                    1$:     MOV     #20,R4  ;THIS PATTERN WILL BE
030004  012702  177750                            MOV     #MAINT,R2           ;PUT IN THE MAINTENANCE
030010  005001                                    CLR     R1                  ;REGISTER
030012  010015                                    MOV     R0,(R5)             ;PUT THE TEST PATTERN IN
                                                                              ;THE TEST ADDRESS
030014  000401                                    BR      64$

        030016                                    LOC=.            ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        030014                                    LOC=-4&LOC
        030020                                    LOC=LOC+4
        030020                                    .=LOC
```

```

                                                                            ;THE REFERENCE TO THIS NEXT INSTRUCTION
                                                                            ;WILL MAKE THE COMPARE INSTRUCTION A HIT
                                                                            ;SO THAT NO SPURIOUS ERROR SHOULD OCCUR
                                                                            ;WHILE THE MAINTENANCE REGISTER IS SET!
030020  010412                          64$:    MOV     R4,(R2)             ;TURN ON THE MAINT. REG.
030022  021500                                  CMP     (R5),R0             ;THE REFERENCE TO (R5)
030024  010112                                  MOV     R1,(R2)             ;SHOULD CAUSE THE ERROR.

030026                                  IIA3:
                                                                            ;THE ERROR DIDN'T OCCUR!
030026  010037  001632                          MOV     R0,$TMP2            ;REPORT FAILURE
030032  012737  030202  001634                  MOV     #IIAT1,$TMP3
030040  005037  001636                          CLR     $TMP4
030044  104144                          64$:    ERROR   144

030046  012737  030244  000114          IIA4:   MOV     #IIAR2,@#CACHVEC            ;SET UP FOR THE GROUP ONE
030054  012737  030046  001512                  MOV     #IIA4,$LPERR        ;ERROR
030062  012737  000044  177746                  MOV     #S1M0,@#CONTRL      ;SELECT GROUP ONE

030070  012705  030202                          MOV     #IIAT1,R5           ;MAKE THE TEST ADDRESS A
030074  005715                                  TST     (R5)                ;HIT, IN GROUP ONE.
030076  005715                                  TST     (R5)

                                                                            ;SEE IF REFERENCE ADDRESS
030100  032737  000010  177752                  BIT     #10,@#HITMIS        ;IS A HIT.
030106  001007                                  BNE     1$
                                                                            ;IF NOT ERROR!
030110  010537  001632                          MOV     R5,$TMP2
030114  012737  000001  001630                  MOV     #1,$TMP1
030122  104001                                  ERROR   1

030124  104420                                  SKIPT                       ;ERROR FATAL. GO TO NEXT TEST.


030126  012704  000100                  1$:     MOV     #100,R4 ;THIS PATTERN WILL BE
030132  012702  177750                          MOV     #MAINT,R2           ;PUT IN THE MAINT. REG.
030136  005001                                  CLR     R1
030140  010015                                  MOV     R0,(R5)             ;PUT THE TEST PATTERN IN (R5),
                                                                            ;IIAT1.
030142  000402                                  BR      50$                 ;PUT THE NEXT INSTRUCTION EXECUTED
                                                                            ;ON AN EVEN WORD BOUNDARY SO THE
                                                                            ;SUBSEQUENT INSTRUCTION, A CMP,
                                                                            ;WILL BE A HIT.

        030144                                  LOC=.                ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        030144                                  LOC=-4&LOC
        030150                                  LOC=LOC+4
        030150                                  .=LOC

030150  000240                          50$:    NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
030152  010412                                  MOV     R4,(R2)             ;TURN ON THE MAINT. REG.
030154  021500                                  CMP     (R5),R0             ;THIS REFERENCE TO (R5) SHOULD
030156  010112                                  MOV     R1,(R2)             ;CAUSE THE ERROR.

030160                                  IIA5:
```

```
                                                                ;THE ERROR DIDN'T OCCUR!
030160  010037  001632                  MOV     R0,$TMP2        ;REPORT FAILURE
030164  012737  030202  001634          MOV     #IIAT1,$TMP3
030172  005037  001636                  CLR     $TMP4
030176  104145                  64$:    ERROR   145

030200  000437                  IIA6:   BR      IIA7

030202  000000                  IIAT1:.WORD     0

030204                          IIAR1:
030204  022737  004500  177744          CMP     #4500,@#MEMERR  ;MAKE SURE THE ERROR
030212  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
030214  000137  054076          1$:     JMP     SPUR
030220  022737  030202  177740  2$:     CMP     #IIAT1,@#LOADRS ;MAKE SURE THE ERROR
030226  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
030230  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
030232  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
030240  000137  030046                  JMP     IIA4            ;GO TEST GROUP ONE
030244                          IIAR2:
030244  022737  004600  177744          CMP     #4600,@#MEMERR  ;MAKE SURE THE ERROR
030252  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
030254  000137  054076          1$:     JMP     SPUR
030260  022737  030202  177740  2$:     CMP     #IIAT1,@#LOADRS ;MAKE SURE THE ERROR
030266  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
030270  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
030272  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.

030300  022700  000377          IIA7:   CMP     #377,R0         ;INCREMENT THE TEST
030304  001404                          BEQ     IIA8            ;PATTERN
030306  062700  000001                  ADD     #1,R0
030312  000137  027702                  JMP     IIA1

030316  104416                  IIA8:   RSET

                                ;;*****************************************************************
                                ;*TEST 31        CACHE DATA MEMORY PARITY CHECKERS HIGH BYTE TEST
                                ;*
                                ;*THIS IS A TEST OF THE TWO CACHE DATA MEMORY PARITY
                                ;*CHECKERS FOR THE HIGH BYTE, ONE FOR EACH GROUP. THE
                                ;*MAINTENANCE REGISTER ISUSED TO FORCE A PARITY A
                                ;*PARITY ERROR AT EVERY DATA PATTERN WHICH HAS A ONE
                                ;*PARITY BIT. NOTE THAT THE CACHE DATA MEMORY PARITY HAS,
                                ;*EFFECTIVELY, ODD PARITY. THE MAINTENANCE FUNCTION ON THE
                                ;*CACHE DATA MEMORY PARITY CHECKERS HAS THE EFFECT OF
                                ;*FORCING THE PARITY BIT OF THE BYTE BEING CHECKED TO
                                ;*ZERO. THIS MEANS THAT ONCE THIS MAINTENANCE FUNCTION
                                ;*IS ENABLED THE ERROR WILL OCCUR ON A SUBSEQUENT
                                ;*READ OF A BYTE WITH A ONE PARITY BIT, THAT IS
                                ;*BYTES WITH ZERO PARITY BITS WILL NOT CAUSE THE ERROR.
                                ;*
                                ;;*****************************************************************
030320  000004                  TST31:  SCOPE
030322  012737  000020  001676          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
```

```
        000032                    IIB=$TN
                                                                  ;SET THE SKAD REGISTER
030330  012737  030774  054230              MOV     #TST32,SKAD      ;IN CASE THE TEST ABORTS.

030336  113737  001502  001626              MOVB    $TSTNM,$TMP0
030344  012737  054076  000114              MOV     #SPUR,@#CACHVEC

030352  005000                              CLR     R0               ;THIS IS THE COUNTER CONTAINING
                                                                     ;THE TEST DATA PATTERN
030354  012737  030354  001512  IIB1:       MOV     #IIB1,$LPERR
030362  004737  054470                      JSR     PC,PARCNT        ;SET IF THIS TEST PATTERN HAS
030366  032702  000001                      BIT     #BIT0,R2         ;THE PARITY BIT SET (1), IF NOT
030372  001402                              BEQ     IIB2             ;GO TO THE NEXT PATTERN
030374  000137  030754                      JMP     IIB7
030400  012737  000030  177746  IIB2:       MOV     #S0M1,@#CONTRL   ;SELECT GROUP ZERO.
030406  012737  030660  000114              MOV     #IIBR1,@#CACHVEC         ;SET UP FOR THE ERROR
030414  012705  030656                      MOV     #IIBT1,R5        ;MAKE THE TEST ADDRESS A
030420  005715                              TST     (R5)             ;HIT IN GROUP ZERO
030422  005715                              TST     (R5)             ;MAKE SURE IT IS A HIT

                                                                     ;SEE IF REFERENCE ADDRESS
030424  032737  000010  177752              BIT     #10,@#HITMIS     ;IS A HIT.
030432  001007                              BNE     1$
                                                                     ;IF NOT ERROR!
030434  010537  001632                      MOV     R5,$TMP2
030440  012737  000000  001630              MOV     #0,$TMP1
030446  104001                              ERROR   1

030450  104420                              SKIPT                    ;ERROR FATAL. GO TO NEXT TEST.


030452  012704  000040          1$:         MOV     #40,R4   ;THIS PATTERN WILL BE
030456  012702  177750                      MOV     #MAINT,R2        ;PUT IN THE MAINTENANCE
030462  005001                              CLR     R1               ;REGISTER
030464  010015                              MOV     R0,(R5)          ;PUT THE TEST PATTERN IN
                                                                     ;THE TEST ADDRESS
030466  000402                              BR      64$

        030470                              LOC=.            ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        030470                              LOC=-4&LOC
        030474                              LOC=LOC+4
        030474                              .=LOC

                                                                     ;THE REFERENCE TO THIS NEXT INSTRUCTION
                                                                     ;WILL MAKE THE COMPARE INSTRUCTION A HIT
                                                                     ;SO THAT NO SPURIOUS ERROR SHOULD OCCUR
                                                                     ;WHILE THE MAINTENANCE REGISTER IS SET!
030474  010412                  64$:        MOV     R4,(R2)          ;TURN ON THE MAINT. REG.
030476  021500                              CMP     (R5),R0          ;THE REFERENCE TO (R5)
030500  010112                              MOV     R1,(R2)          ;SHOULD CAUSE THE ERROR.

030502                          IIB3:
                                                                     ;THE ERROR DIDN'T OCCUR!
030502  010037  001632                      MOV     R0,$TMP2         ;REPORT FAILURE
030506  012737  030656  001634              MOV     #IIBT1,$TMP3
030514  005037  001636                      CLR     $TMP4
```

```
030520  104146                          64$:    ERROR   146

030522  012737  030720  000114          IIB4:   MOV     #IIBR2,@#CACHVEC         ;SET UP FOR THE GROUP ONE
030530  012737  030522  001512                  MOV     #IIB4,$LPERR    ;ERROR
030536  012737  000044  177746                  MOV     #S1M0,@#CONTRL  ;SELECT GROUP ONE

030544  012705  030656                          MOV     #IIBT1,R5       ;MAKE THE TEST ADDRESS A
030550  005715                                  TST     (R5)            ;HIT, IN GROUP ONE.
030552  005715                                  TST     (R5)


                                                                        ;SEE IF REFERENCE ADDRESS
030554  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
030562  001007                                  BNE     1$
                                                                        ;IF NOT ERROR!
030564  010537  001632                          MOV     R5,$TMP2
030570  012737  000001  001630                  MOV     #1,$TMP1
030576  104001                                  ERROR   1

030600  104420                                  SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.


030602  012704  000200                  1$:     MOV     #200,R4 ;THIS PATTERN WILL BE
030606  012702  177750                          MOV     #MAINT,R2       ;PUT IN THE MAINT. REG.
030612  005001                                  CLR     R1
030614  010015                                  MOV     R0,(R5)         ;PUT THE TEST PATTERN IN (R5),
                                                                        ;IIBT1.
030616  000402                                  BR      50$             ;PUT THE NEXT INSTRUCTION EXECUTED
                                                                        ;ON AN EVEN WORD BOUNDARY SO THE
                                                                        ;SUBSEQUENT INSTRUCTION, A CMP,
                                                                        ;WILL BE A HIT.

        030620                                  LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        030620                                  LOC=-4&LOC
        030624                                  LOC=LOC+4
        030624                                  .=LOC

030624  000240                          50$:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
030626  010412                                  MOV     R4,(R2)         ;TURN ON THE MAINT. REG.
030630  021500                                  CMP     (R5),R0         ;THIS REFERENCE TO (R5) SHOULD
030632  010112                                  MOV     R1,(R2)         ;CAUSE THE ERROR.

030634                                  IIB5:
                                                                        ;THE ERROR DIDN'T OCCUR!
030634  010037  001632                          MOV     R0,$TMP2        ;REPORT FAILURE
030640  012737  030656  001634                  MOV     #IIBT1,$TMP3
030646  005037  001636                          CLR     $TMP4
030652  104147                          64$:    ERROR   147

030654  000437                          IIB6:   BR      IIB7

030656  000000                          IIBT1:.WORD     0

030660                                  IIBR1:
030660  022737  004500  177744                  CMP     #4500,@#MEMERR  ;MAKE SURE THE ERROR
030666  001402                                  BEQ     2$              ;REGISTER IS SET PROPERLY
030670  000137  054076                  1$:     JMP     SPUR
```

```
030674  022737  030656  177740  2$:     CMP     #IIBT1,@#LOADRS ;MAKE SURE THE ERROR
030702  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
030704  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
030706  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
030714  000137  030522                  JMP     IIB4            ;GO TEST GROUP ONE
030720                          IIBR2:
030720  022737  004600  177744          CMP     #4600,@#MEMERR  ;MAKE SURE THE ERROR
030726  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
030730  000137  054076          1$:     JMP     SPUR
030734  022737  030656  177740  2$:     CMP     #IIBT1,@#LOADRS ;MAKE SURE THE ERROR
030742  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
030744  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
030746  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.

030754  022700  177400          IIB7:   CMP     #177400,R0                ;INCREMENT THE TEST
030760  001404                          BEQ     IIB8            ;PATTERN
030762  062700  000400                  ADD     #400,R0
030766  000137  030354                  JMP     IIB1

030772  104416                  IIB8:   RSET



                                ;;*****************************************************************
                                ;*TEST 32        CACHE DATA MEMORY WORST CASE NOISE TEST
                                ;*
                                ;*THIS TEST DOES A GALLOPING 0'S AND 1'S OR PING PONG
                                ;*TEST ON THE CACHE BIPOLAR DATA MEMORY.
                                ;*
                                ;;*****************************************************************
030774  000004                  TST32:  SCOPE
                                                                ;SET THE SKAD REGISTER
030776  012737  032132  054230          MOV     #TST33,SKAD     ;IN CASE THE TEST ABORTS.


031004  012737  054076  000114          MOV     #SPUR,@#CACHVEC
031012  113737  001502  001626          MOVB    $TSTNM,$TMPO    ;SAVE TESTN FOR PRINT OUT.

031020  005037  031522                  CLR     QQPAT1          ;BACK ROUND PATTERN OF
                                                                ;0'S FOR THE GALLOPING
                                                                ;1'S TEST TO BE EXECUTED
                                                                ;FIRST.
031024  012737  000001  031516          MOV     #1,QQFLG2       ;QQFLG=1 MEANS GALLOPING
                                                                ;ONES TEST IN PROGRESS.
                                                                ;QQFLG=0 MEANS GALLOPING
                                                                ;ZEROES TEST IN PROGRESS.
031032  012737  031032  001512  QQ1:    MOV     #QQ1,$LPERR     ;SET ERROR LOOP INITIALLY
                                                                ;TO THIS POINT.
031040  012737  000044  031532          MOV     #S1M0,QQGS      ;TEST GROUP 1 FIRST.
031046  012737  000030  031534          MOV     #S0M1,QQGM      ;S0M1 AND S1M0 ARE
                                                                ;PATTERNS WHICH WILL BE
                                                                ;LOADED INTO THE CACHE
                                                                ;CONTROL REGISTER TO
                                                                ;(SELECT GRP0 * MISS GRP1)
                                                                ;AND (SELECT GRP1 * MISS GRP0)
```

```
                                                                  ;RESPECTIVELY.
031054  012737  000001  031520          MOV     #1,QQFLG1         ;QQFLG ONE CONTAINS THE
                                                                  ;NUMBER OF THE GROUP
                                                                  ;BEING TESTED, INITIALLY 1.

031062  012703  031062          QQ2:    MOV     #QQ2,R3           ;MAKE LOCATIONS QQ1
031066  012704  001000                  MOV     #1000,R4          ;THROUGH QQ2 + 2000 (OCT)
031072  013737  031534  177746  1$:     MOV     QQGM,@#CONTRL     ;HITS IN THE GROUP NOT
031100  005713                          TST     (R3)              ;BEING TESTED WHILE
031102  013737  031532  177746          MOV     QQGS,@#CONTRL     ;GETTING THESE LOCATIONS
031110  005763  002000                  TST     2000(R3)          ;TO BE MISSES IN THE
031114  062703  000002                  ADD     #2,R3             ;GROUP THAT IS BEING
031120  077414                          SOB     R4,1$             ;TESTED
031122  012704  001000                  MOV     #1000,R4          ;MAKE LOCATIONS TESTR2
031126  012705  142000                  MOV     #TESTR2,R5        ;THROUGH TESTR2+2000(OCT)
031132  013703  031532                  MOV     QQGS,R3           ;HITS IN THE GROUP
031136  042703  177717                  BIC     #177717,R3        ;BEING TESTED WHILE
031142  010337  177746                  MOV     R3,@#CONTRL       ;WRITING THE BACKGROUND
031146  013715  031522          QQ3:    MOV     QQPAT1,(R5)       ;PATTERN, IN QQPAT1, IN
031152  005715                          TST     (R5)
031154  005725                          TST     (R5)+             ;THEM. MAKE SURE THEY
031156  032737  000010  177752          BIT     #10,@#HITMIS      ;ARE HITS
031164  001011                          BNE     QQ4
031166  013737  031520  001630          MOV     QQFLG1,$TMP1      ;IF NOT ERROR
031174  010537  001632                  MOV     R5,$TMP2
031200  062737  177776  001632          ADD     #-2,$TMP2
031206  104001                  1$:     ERROR   1
031210  077422                  QQ4:    SOB     R4,QQ3
031212  013703  031534                  MOV     QQGM,R3           ;FROM NOW ON SELECT
031216  042703  177717                  BIC     #177717,R3        ;THE GROUP NOT BEING
031222  010337  177746                  MOV     R3,@#CONTRL       ;TESTED

031226  012704  031536                  MOV     #QQ10,R4          ;THE THREE ROUTINES
031232  042704  176000                  BIC     #176000,R4        ;QQ10-QQ11, QQ12-QQ13 AND
031236  012705  031612                  MOV     #QQ11,R5          ;QQ14-QQ15 ARE IDENTICAL
031242  042705  176000                  BIC     #176000,R5        ;EXCEPT FOR WHAT PART
031246  020405                          CMP     R4,R5             ;OF THE CACHE GROUP THAT
031250  002407                          BLT     QQ5               ;IS NOT BEING TEST THEY
031252  012737  031614  031512          MOV     #QQ12,QQLO        ;LIE IN. HERE DECIDE
031260  012737  031672  031514          MOV     #QQ14,QQHI        ;WHICH TWO OF THE
031266  000450                          BR      QQ8               ;ABOVE THREE IS APPROPRIATE
031270  012704  031614          QQ5:    MOV     #QQ12,R4          ;FOR THIS TEST.
031274  042704  176000                  BIC     #176000,R4
031300  012705  031670                  MOV     #QQ13,R5
031304  042705  176000                  BIC     #176000,R5
031310  020405                          CMP     R4,R5
031312  002407                          BLT     QQ6
031314  013737  031672  031512          MOV     QQ14,QQLO
031322  013737  031536  031514          MOV     QQ10,QQHI
031330  000427                          BR      QQ8
031332  013704  031536          QQ6:    MOV     QQ10,R4
031336  042704  176000                  BIC     #176000,R4
031342  012705  031614                  MOV     #QQ12,R5
031346  042705  176000                  BIC     #176000,R5
031352  020405                          CMP     R4,R5
031354  003007                          BGT     QQ7
```

```
031356  012737  031536  031512            MOV     #QQ10,QQLO
031364  012737  031614  031514            MOV     #QQ12,QQHI
031372  000406                            BR      QQ8
031374  012737  031614  031512  QQ7:      MOV     #QQ12,QQLO
031402  012737  031536  031514            MOV     #QQ10,QQHI

031410  012702  142000          QQ8:      MOV     #TESTR2,R2          ;INITIALIZE FOR EITHER
031414  012701  140000                    MOV     #TESTR1,R1          ;THE GALLOPING ONES OR
031420  012705  001000                    MOV     #1000,R5            ;GALLOPING ZEROES TEST
                                                                      ;WHICH IS PENDING.
031424  012737  032034  000114            MOV     #QQERR1,@#CACHVEC         ;IF THE TEST FAILS A
                                                                      ;PARITY ABORT IS LIKELY
                                                                      ;SO SET UP TO GO THE
                                                                      ;ERROR ROUTINE.
031432  012737  031440  001512            MOV     #QQ9,$LPERR         ;SET THE LOOP ERROR
                                                                      ;ADDRESS FOR THE BEGINNING
                                                                      ;OF THE PASS ROUTINE.

031440  012703  142000          QQ9:      MOV     #TESTR2,R3          ;THIS DOES ONE PASS OF
031444  012704  001000                    MOV     #1000,R4            ;THE TEST FOR EACH LOCATION.
031450  005112                            COM     (R2)                ;PUT THE GALLOPING PATTERN
                                                                      ;IN THE MEMORY.

031452  010100                  QQ9.5:    MOV     R1,R0               ;SEE WHICH OF THE
031454  042700  176000                    BIC     #176000,R0          ;TWO ROUTINES (QQ10,QQ12 OR
031460  013737  031514  031524            MOV     QQHI,QQTMP1         ;QQ14) SHOULD FINISH
031466  042737  176000  031524            BIC     #176000,QQTMP1      ;SETTING FOR THIS TEST
031474  020037  031524                    CMP     R0,QQTMP1           ;PASS.
031500  002402                            BLT     1$
031502  000177  000004                    JMP     @QQLO
031506  000177  000002          1$:       JMP     @QQHI

031512  000000                  QQLO:     .WORD   0                   ;QQLO AND QQHI CONTAIN THE
031514  000000                  QQHI:     .WORD   0                   ;ADDRESSES OF THE ROUTINES
                                                                      ;TO BE USED IN SETTING UP
                                                                      ;FOR A PASS.
031516  000000                  QQFLG2:  .WORD    0                   ;1 IF DOING GALLOPING 1'S TEST.
                                                                      ;0 IF DOING GALLOPING 0'S TEST.
031520  000000                  QQFLG1:  .WORD    0                   ;GROUP BEING TESTED, 1 OR 0.
031522  000000                  QQPAT1:  .WORD    0                   ;0 OR 1 BACKGROUND PATTERN.
031524  000000                  QQTMP1:  .WORD    0                   ;USED AS TEMPORARY STORAGE.
031526  000000                  QQTMP2:  .WORD    0
031530  000000                  QQTMP3:  .WORD    0
031532  000000                  QQGS:     .WORD   0                   ;THESE REGISTERS HOLD PATTERNS
031534  000000                  QQGM:     .WORD   0                   ;WHICH ARE TO BE LOADED INTO THE
                                                                      ;CACHE CONTROL REGISTER.

                                ;THIS ROUTINE IS USED TO SET UP THE INSTRUCTIONS:
                                ;         1$:     CMP     (R3)+,(R2)
                                ;                 SOB     R4,1$
                                ;                 JMP     @#QQ16
                                ;IN POSITION, AS HITS IN THE GROUP NOT BEING TESTED.
031536  000240                  QQ10:     NOP                         ;USED AS A BUFFER SO
031540  000240                            NOP                         ;THIS CODE WON'T WIPE
                                                                      ;OUT DESIRED HITS
031542  012711  022312                    MOV     #022312,(R1)        ;020323=(CMP (R3)+,(R2)
031546  005711                                   TST     (R1)
031550  012761  077402  000002                   MOV     #077402,2(R1)    ;077402=(SOB R4,.-2)
031556  005761  000002                           TST     2(R1)
031562  012761  000137  000004                   MOV     #000137,4(R1)    ;000137=(JMP @#QQ16)
031570  005761  000004                           TST     4(R1)            ;QQ16
031574  012761  031750  000006                   MOV     #QQ16,6(R1)
031602  005761  000006                           TST     6(R1)
031606  000111                                   JMP     (R1)             ;GO DO A PASS.
031610  000240                                   NOP
031612  000240                          QQ11:    NOP

                                        ;THIS ROUTINE IS USED TO SET UP THE INSTRUCTIONS:
                                        ;        1$:      CMP     (R3)+,(R2)
                                        ;                 SOB     R4,1$
                                        ;                 JMP     @#QQ16
                                        ;IN POSITION, AS HITS IN THE GROUP NOT BEING TESTED.
031614  000240                          QQ12:    NOP                      ;USED AS A BUFFER SO
031616  000240                                   NOP                      ;THIS CODE WON'T WIPE
                                                                          ;OUT DESIRED HITS
031620  012711  022312                           MOV     #022312,(R1)     ;020323=(CMP (R3)+,(R2)
031624  005711                                   TST     (R1)
031626  012761  077402  000002                   MOV     #077402,2(R1)    ;077402=(SOB R4,.-2)
031634  005761  000002                           TST     2(R1)
031640  012761  000137  000004                   MOV     #000137,4(R1)    ;000137=(JMP @#QQ16)
031646  005761  000004                           TST     4(R1)            ;QQ16
031652  012761  031750  000006                   MOV     #QQ16,6(R1)
031660  005761  000006                           TST     6(R1)
031664  000111                                   JMP     (R1)             ;GO DO A PASS.
031666  000240                                   NOP
031670  000240                          QQ13:    NOP

                                        ;THIS ROUTINE IS USED TO SET UP THE INSTRUCTIONS:
                                        ;        1$:      CMP     (R3)+,(R2)
                                        ;                 SOB     R4,1$
                                        ;                 JMP     @#QQ16
                                        ;IN POSITION, AS HITS IN THE GROUP NOT BEING TESTED.
031672  000240                          QQ14:    NOP                      ;USED AS A BUFFER SO
031674  000240                                   NOP                      ;THIS CODE WON'T WIPE
                                                                          ;OUT DESIRED HITS
031676  012711  022312                           MOV     #022312,(R1)     ;020323=(CMP (R3)+,(R2)
031702  005711                                   TST     (R1)
031704  012761  077402  000002                   MOV     #077402,2(R1)    ;077402=(SOB R4,.-2)
031712  005761  000002                           TST     2(R1)
031716  012761  000137  000004                   MOV     #000137,4(R1)    ;000137=(JMP @#QQ16)
031724  005761  000004                           TST     4(R1)            ;QQ16
031730  012761  031750  000006                   MOV     #QQ16,6(R1)
031736  005761  000006                           TST     6(R1)
031742  000111                                   JMP     (R1)             ;GO DO A PASS.
031744  000240                                   NOP
031746  000240                          QQ15:    NOP

031750  005122                          QQ16:    COM     (R2)+            ;PASS DONE. RESTORE THE
                                                                          ;BACKGROUND PATTERN.

031752  062701  000002                  QQ17:    ADD     #2,R1            ;GO TO NEXT LOCATION FOR
                                                                          ;NEXT PASS.
```

```
031756  005305                                  DEC     R5              ;DO ANOTHER PASS?
031760  001402                                  BEQ     1$
031762  000137  031440                          JMP     QQ9
031766                                  1$:
031766  012737  000044  031534                  MOV     #S1M0,QQGM      ;TESTED GROUP 1 NOW GO BACK
031774  012737  000030  031532                  MOV     #S0M1,QQGS      ;AND TEST GROUP 0
032002  005337  031520                          DEC     QQFLG1
032006  001002                                  BNE     QQ18
032010  000137  031062                          JMP     QQ2

032014  012737  177777  031522  QQ18:   MOV     #-1,QQPAT1      ;GALLOPING 1'S TEST IS
032022  005337  031516                          DEC     QQFLG2          ;COMPLETE, ON BOTH GROUPS,
032026  001041                                  BNE     QQ19            ;SET THE BACKGROUND PATTERN
032030  000137  031032                          JMP     QQ1             ;FOR GALLOPING 0'S AND GO
                                                                        ;BACK TO PERFORM THIS TEST
                                                                        ;ON BOTH GROUPS.

032034  013737  177744  001630  QQERR1: MOV     @#MEMERR,$TMP1  ;COME HERE IF DURING THE
032042  013737  177740  001632          MOV     @#LOADRS,$TMP2  ;TEST A TRAP OR ABORT
032050  013737  177742  001634          MOV     @#HIADRS,$TMP3  ;OCCURRED TO CACHVEC
032056  011637  001636                          MOV     (SP),$TMP4
032062  022626                                  CMP     (SP)+,(SP)+
032064  010137  001640                          MOV     R1,$TMP5
032070  013737  031520  001642                  MOV     QQFLG1,$TMP6
032076  032737  000600  001630                  BIT     #600,$TMP1
032104  001002                                  BNE     QQERR2
032106  104002                                  ERROR   2
032110  000406                                  BR      QQERR4
032112  005737  031522          QQERR2: TST     QQPAT1          ;GALLOPING 1' OR 0'S?
032116  001002                                  BNE     QQERR3
032120  104003                                  ERROR   3               ;0'S.
032122  000401                                  BR      QQERR4
032124  104004                          QQERR3: ERROR   4               ;1'S
032126  000137  031750          QQERR4: JMP     QQ16            ;CONTINUE?

032132                                  QQ19:                           ;DONE! PERHAPS PRINT SUMMARY.
                                                                        ;?????


                                        ;;*****************************************************************
                                        ;*TEST 33       CACHE DATA MEMORY CHIP SELECTION LOGIC TEST
                                        ;*
                                        ;*THIS ROUTINE TESTS THE 'CHIP-SET' ENABLE LOGIC FOR THE CACHE DATA
                                        ;*MEMORY.  TO DEFINE THE TERM 'CHIP-SET' CONSIDER THE CACHE MEMORY AS
                                        ;*BEING DIVIDED INTO FOUR SETS OF 256 (DEC) X 1 BIT BIPOLAR MEMORY
                                        ;*CHIPS.  EACH SET IS MADE UP OF 18 CHIPS, THE 74S200, EACH CHIP
                                        ;*REPRESENTS ONE BIT OF DATA OR PARITY, THUS 16 DATA BITS PLUS
                                        ;*TWO PARITY BITS CORRESPOND TO THE 18 CHIPS IN EACH GROUP.
                                        ;*THE 'CHIP-SETS' THEN CORRESPOND TO THE STRUCTURE OF THE MEMORY
                                        ;*IN THIS WAY:
                                        ;*      SET 0   GROUP 0 EVEN WORD
                                        ;*      SET 1   GROUP 0 ODD WORD
                                        ;*      SET 2   GROUP 1 EVEN WORD
                                        ;*      SET 3   GROUP 1 ODD WORD
                                        ;*A DIFFERENT PATTERN, 000000 177777 125252 AND 052525, IS WRITTEN
                                        ;*INTO EACH GROUP AND THEN READ BACK.  EVERY PERMUTATION OF THE
                                        ;*FOUR TEST PATTERNS IN THE FOUR SETS IS TRIED AND CHECKED.
                                        ;*FOR EACH PERMUTATION OF THE TEST PATTERNS THIS ROUTINE FIRST WRITES
                                        ;*'UP' (SET 0 FIRST THEN 1,2 AND 3) THEN 'DOWN' (SET 3 FIRST THEN 2,1 AND 0).
                                        ;*
                                        ;;*****************************************************************
032132  000004                          TST33:  SCOPE
032134  012737  000040  001676                  MOV     #40,$TIMES          ;;DO 40 ITERATIONS
                                                                            ;SET THE SKAD REGISTER
032142  012737  033676  054230                  MOV     #TST34,SKAD         ;IN CASE THE TEST ABORTS.


032150  113737  001502  001626                  MOVB    $TSTNM,$TMPO        ;PUT THE TEST NUMBER IN
                                                                            ;$TMPO FOR PRINT OUT.
032156  012737  054076  000114                  MOV     #SPUR,@#CACHVEC     ;EXPECT NO PARITY ERRORS.

032164  012737  000014  177746          KK1:    MOV     #MOM1,@#CONTRL      ;FORCE MISSES AND
032172  005037  033532                          CLR     KKPAT1              ;INITIALIZE THE TEST PATTERN
032176  012737  177777  033534                  MOV     #177777,KKPAT2      ;TABLE
032204  012737  125252  033536                  MOV     #125252,KKPAT3
032212  012737  052525  033540                  MOV     #52525,KKPAT4

032220  005037  033526                          CLR     KKFLG1              ;INITIALIZE KKFLG1:
                                                                            ;0 MEANS WRITE PATTERNS IN
                                                                            ;IN THE UPWARD DIRECTION
                                                                            ;1 MEANS WRITE PATTERNS IN
                                                                            ;THE DOWNWARD DIRECTION

032224  012700  033546                  KK2:    MOV     #KKTMP2,R0          ;ESTABLISH AN OFFSET FOR
032230  042700  176003                          BIC     #176003,R0          ;A TEST AREA WHOSE HITS
                                                                            ;WILL NOT BE INTERFERRED WITH BY
032234  010001                                  MOV     R0,R1               ;THE CYCLES CAUSED WHILE
032236  062701  140000                          ADD     #TESTR1,R1          ;FETCHING THE TEST CODE.
032242  010002                                  MOV     R0,R2
032244  062702  142000                          ADD     #TESTR2,R2

032250  010137  001640                          MOV     R1,$TMP5            ;SAVE THE ADDRESSES OF
032254  010137  001642                          MOV     R1,$TMP6            ;THE FOUR TEST WORD LOCATIONS,
032260  062737  000002  001642                  ADD     #2,$TMP6            ;FOR TYPE OUT IN CASE
032266  010237  001644                          MOV     R2,$TMP7            ;OF ERROR.
032272  010237  001646                          MOV     R2,$TMP10
032276  062737  000002  001646                  ADD     #2,$TMP10

032304  012705  033534                          MOV     #KKPAT2,R5          ;A POINTER USED IN GENERATING
                                                                            ;EVERY PERMUTATION OF THE TEST
                                                                            ;PATTERNS.
032310  012700  000006                          MOV     #6,R0               ;R0 AND KKCNT1 ARE ALSO USED
032314  012737  000004  033530                  MOV     #4,KKCNT1           ;IN GENERATING THE PERMUTATIONS.

032322  012737  032330  001512                  MOV     #KK3,$LPERR         ;WHEN LOOPING ON ERROR GO TO KK3.
032330  000240                          KK3:    NOP                         ;FOR SCOPING PER POSES
032332  012737  000034  177746                  MOV     #S0MOM1,@#CONTRL;MAKE THE TEST AREA HITS
032340  005711                                  TST     (R1)                ;IN THE CACHE GROUPS.
032342  005761  000002                          TST     2(R1)
032346  012737  000054  177746                  MOV     #S1MOM1,@#CONTRL
032354  005712                                  TST     (R2)
032356  005762  000002                          TST     2(R2)
032362  005037  177746                          CLR     @#CONTRL
```

```


032366  005711                                  TST     (R1)

                                                                        ;SEE IF REFERENCE ADDRESS
032370  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
032376  001006                                  BNE     1$
                                                                        ;IF NOT ERROR!
032400  010137  001632                          MOV     R1,$TMP2
032404  012737  000000  001630                  MOV     #0,$TMP1
032412  104001                                  ERROR   1



032414                          1$:

032414  005761  000002                          TST     2(R1)

                                                                        ;SEE IF REFERENCE ADDRESS
032420  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
032426  001011                                  BNE     2$
                                                                        ;IF NOT ERROR!
032430  010137  001632                          MOV     R1,$TMP2
032434  062737  000002  001632                  ADD     #2,$TMP2
032442  012737  000000  001630                  MOV     #0,$TMP1
032450  104001                                  ERROR   1



032452                          2$:

032452  005712                                  TST     (R2)

                                                                        ;SEE IF REFERENCE ADDRESS
032454  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
032462  001006                                  BNE     3$
                                                                        ;IF NOT ERROR!
032464  010237  001632                          MOV     R2,$TMP2
032470  012737  000001  001630                  MOV     #1,$TMP1
032476  104001                                  ERROR   1



032500                          3$:

032500  005762  000002                          TST     2(R2)

                                                                        ;SEE IF REFERENCE ADDRESS
032504  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
032512  001011                                  BNE     4$
                                                                        ;IF NOT ERROR!
032514  010237  001632                          MOV     R2,$TMP2
032520  062737  000002  001632                  ADD     #2,$TMP2
032526  012737  000001  001630                  MOV     #1,$TMP1
032534  104001                                  ERROR   1

```

```



032536  005737  033526          4$:     TST     KKFLG1            ;SEE IF THE TST PATTERN
                                                                  ;SHOULD BE WRITTEN UPWARD
                                                                  ;OR DOWNWARD.
032542  001045                          BNE     KK4               ;BRANCH IF DOWNWARD
                                                                  ;OTHERWISE WRITE IT IN THE
                                                                  ;UPWARD DIRECTION.
032544  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032552  013703  033532                  MOV     KKPAT1,R3                   ;LOCATION KKPAT1, INTO THE
032556  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R1 PLUS 0
032562  010361  000000                  MOV     R3,0(R1)
032566  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032574  013703  033534                  MOV     KKPAT2,R3                   ;LOCATION KKPAT2, INTO THE
032600  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R1 PLUS 2
032604  010361  000002                  MOV     R3,2(R1)
032610  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032616  013703  033536                  MOV     KKPAT3,R3                   ;LOCATION KKPAT3, INTO THE
032622  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R2 PLUS 0
032626  010362  000000                  MOV     R3,0(R2)
032632  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032640  013703  033540                  MOV     KKPAT4,R3                   ;LOCATION KKPAT4, INTO THE
032644  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R2 PLUS 2
032650  010362  000002                  MOV     R3,2(R2)
032654  000444                          BR      KK5
032656                          KK4:                              ;WRITE THE PATTERN IN THE
                                                                  ;DOWNWARD DIRECTION
032656  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032664  013703  033540                  MOV     KKPAT4,R3                   ;LOCATION KKPAT4, INTO THE
032670  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R2 PLUS 2
032674  010362  000002                  MOV     R3,2(R2)
032700  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032706  013703  033536                  MOV     KKPAT3,R3                   ;LOCATION KKPAT3, INTO THE
032712  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R2 PLUS 0
032716  010362  000000                  MOV     R3,0(R2)
032722  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032730  013703  033534                  MOV     KKPAT2,R3                   ;LOCATION KKPAT2, INTO THE
032734  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R1 PLUS 2
032740  010361  000002                  MOV     R3,2(R1)
032744  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;WRITE THE TEST PATTERN, FROM
032752  013703  033532                  MOV     KKPAT1,R3                   ;LOCATION KKPAT1, INTO THE
032756  005037  177746                  CLR     @#CONTRL          ;ADDRESS IN R1 PLUS 0
032762  010361  000000                  MOV     R3,0(R1)

032766                          KK5:
032766  012737  000014  177746          MOV     #MOM1,@#CONTRL
032774  013703  033532                  MOV     KKPAT1,R3                   ;SEE IF THE TEST PATTERN WAS
033000  005037  177746                  CLR     @#CONTRL          ;WRITTEN OR IS READ CORRECTLY.
033004  016104  000000                  MOV     0(R1),R4

                                                                  ;SEE IF REFERENCE ADDRESS
033010  032737  000010  177752          BIT     #10,@#HITMIS      ;IS A HIT.
033016  001006                          BNE     64$
                                                                  ;IF NOT ERROR!
033020  010137  001632                  MOV     R1,$TMP2
```

```
033024  012737  000000  001630            MOV     #0,$TMP1
033032  104001                            ERROR   1


033034  020403                    64$:    CMP     R4,R3
033036  001402                            BEQ     KK6
033040  004737  033556                    JSR     PC,KKERR1

033044                            KK6:
033044  012737  000014  177746            MOV     #MOM1,@#CONTRL
033052  013703  033534                    MOV     KKPAT2,R3               ;SEE IF THE TEST PATTERN WAS
033056  005037  177746                    CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
033062  016104  000002                    MOV     2(R1),R4

                                                                  ;SEE IF REFERENCE ADDRESS
033066  032737  000010  177752            BIT     #10,@#HITMIS    ;IS A HIT.
033074  001011                            BNE     64$
                                                                  ;IF NOT ERROR!
033076  010137  001632                    MOV     R1,$TMP2
033102  062737  000002  001632            ADD     #2,$TMP2
033110  012737  000000  001630            MOV     #0,$TMP1
033116  104001                            ERROR   1


033120  020403                    64$:    CMP     R4,R3
033122  001402                            BEQ     KK7
033124  004737  033570                    JSR     PC,KKERR2

033130                            KK7:
033130  012737  000014  177746            MOV     #MOM1,@#CONTRL
033136  013703  033536                    MOV     KKPAT3,R3               ;SEE IF THE TEST PATTERN WAS
033142  005037  177746                    CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
033146  016204  000000                    MOV     0(R2),R4

                                                                  ;SEE IF REFERENCE ADDRESS
033152  032737  000010  177752            BIT     #10,@#HITMIS    ;IS A HIT.
033160  001006                            BNE     64$
                                                                  ;IF NOT ERROR!
033162  010237  001632                    MOV     R2,$TMP2
033166  012737  000001  001630            MOV     #1,$TMP1
033174  104001                            ERROR   1


033176  020403                    64$:    CMP     R4,R3
033200  001402                            BEQ     KK8
033202  004737  033610                    JSR     PC,KKERR3

033206                            KK8:
033206  012737  000014  177746            MOV     #MOM1,@#CONTRL
033214  013703  033540                    MOV     KKPAT4,R3               ;SEE IF THE TEST PATTERN WAS
033220  005037  177746                    CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
033224  016204  000002                    MOV     2(R2),R4

                                                                  ;SEE IF REFERENCE ADDRESS
033230  032737  000010  177752            BIT     #10,@#HITMIS    ;IS A HIT.
033236  001011                            BNE     64$
```

```
                                                                        ;IF NOT ERROR!
033240  010237  001632                          MOV     R2,$TMP2
033244  062737  000002  001632                  ADD     #2,$TMP2
033252  012737  000001  001630                  MOV     #1,$TMP1
033260  104001                                  ERROR   1


033262  020403                          64$:    CMP     R4,R3
033264  001402                                  BEQ     KK10
033266  004737  033624                          JSR     PC,KKERR4

033272  005737  033526                  KK10:   TST     KKFLG1          ;SEE IF THIS PERMUTATION OF
033276  001005                                  BNE     KK11            ;THE TEST PATTERN HAS BEEN
033300  012737  177777  033526                  MOV     #-1,KKFLG1      ;WRITTEN BOTH UPWARD AND
033306  000137  032330                          JMP     KK3             ;DOWNWARD. IF NOT, KKFLG IS 0,
                                                                        ;GO BACK TO WRITE IT DOWNWARD.

033312  005037  033526                  KK11:   CLR     KKFLG1          ;GENERATE THE NEXT PERMUTATION
033316  012737  000014  177746                  MOV     #MOM1,@#CONTRL  ;OF THE TEST PATTERN IN THE
                                                                        ;TEST TABLE
033324  020527  033540                          CMP     R5,#KKPAT4
033330  001011                                  BNE     KK12

033332  011537  033542                          MOV     (R5),KKPAT5
033336  013715  033534                          MOV     KKPAT2,(R5)
033342  012705  033534                          MOV     #KKPAT2,R5
033346  013715  033542                          MOV     KKPAT5,(R5)
033352  000406                                  BR      KK13

033354  012537  033542                  KK12:   MOV     (R5)+,KKPAT5
033360  011565  177776                          MOV     (R5),-2(R5)
033364  013715  033542                          MOV     KKPAT5,(R5)

033370  005300                          KK13:   DEC     R0
033372  001402                                  BEQ     KK14
033374  000137  032330                          JMP     KK3             ;GO DO NEXT PERMUTATION.

033400  012700  000006                  KK14:   MOV     #6,R0
033404  013737  033532  033542                  MOV     KKPAT1,KKPAT5
033412  005337  033530                          DEC     KKCNT1

033416  022737  000003  033530                  CMP     #3,KKCNT1
033424  001010                                  BNE     KK15

033426  013737  033534  033532                  MOV     KKPAT2,KKPAT1
033434  013737  033542  033534                  MOV     KKPAT5,KKPAT2
033442  000137  032330                          JMP     KK3             ;GO DO NEXT PERMUTATION.

033446  022737  000002  033530          KK15:   CMP     #2,KKCNT1
033454  001010                                  BNE     KK16

033456  013737  033536  033532                  MOV     KKPAT3,KKPAT1
033464  013737  033542  033536                  MOV     KKPAT5,KKPAT3
033472  000137  032330                          JMP     KK3             ;GO DO NEXT PERMUTION.

033476  022737  000001  033530          KK16:   CMP     #1,KKCNT1
```

```
033504  001073                                  BNE     KK17                    ;BRANCH IF DONE!

033506  013737  033540  033532                  MOV     KKPAT4,KKPAT1
033514  013737  033542  033540                  MOV     KKPAT5,KKPAT4
033522  000137  032330                          JMP     KK3                     ;GO DO NEXT PERMUTATION.


033526  000000                          KKFLG1: .WORD   0                       ;0 IF STORING PATTERN UPWARD
                                                                                ;1 IF STORING DOWNWARD.

033530  000000                          KKCNT1: .WORD   0                       ;COUNTER USED IN GENERATING
                                                                                ;THE TEST PATTERN PERMUTATIONS.

033532  000000                          KKPAT1: .WORD   0                       ;TEST PATTERN TABLE.
033534  000000                          KKPAT2: .WORD   0
033536  000000                          KKPAT3: .WORD   0
033540  000000                          KKPAT4: .WORD   0
033542  000000                          KKPAT5: .WORD   0

033544  000000                          KKTMP1: .WORD   0                       ;USED TO LOCATE A TEST AREA WHOSE
033546  000000  000000  000000          KKTMP2: .WORD   0,0,0,0                 ;HITS WON'T BE WIPED OUT BY TEST CODE.
033554  000000

033556  010137  001636                  KKERR1: MOV     R1,$TMP4                ;ERROR REPORTING ROUTINES
033562  005037  001634                          CLR     $TMP3
033566  000427                                  BR      KKERR5

033570  010137  001636                  KKERR2: MOV     R1,$TMP4
033574  062737  000002  001636                  ADD     #2,$TMP4
033602  005037  001634                          CLR     $TMP3
033606  000417                                  BR      KKERR5

033610  010237  001636                  KKERR3: MOV     R2,$TMP4
033614  013737  000001  001634                  MOV     1,$TMP3
033622  000411                                  BR      KKERR5

033624  010237  001636                  KKERR4: MOV     R2,$TMP4
033630  062737  000002  001636                  ADD     #2,$TMP4
033636  012737  000001  001634                  MOV     #1,$TMP3
033644  000400                                  BR      KKERR5

033646  010337  001632                  KKERR5: MOV     R3,$TMP2
033652  011637  001630                          MOV     (SP),$TMP1
033656  012737  000014  177746                  MOV     #MOM1,@#CONTRL

033664  104021                                  ERROR   21

033666  005037  177746                          CLR     @#CONTRL
033672  000207                                  RTS     PC

033674  104416                          KK17:   RSET                            ;DONE!

                                        ;;******************************************************************
                                        ;*TEST 34       CACHE DATA MEMORY BYTE ENABLE LOGIC TEST
                                        ;*
                                        ;*THIS TEST PERFORMS A CHECK OF THE BYTE ENABLE LOGIC
                                        ;*IN THE CACHE DATA MEMORY. THE BYTE PATTERNS 1, 2, 4, 10, 20,
                                        ;*40, 100 A 200 ARE USED. THE FIRST FOUR PATTERNS ARE WRITTEN
                                        ;*IN CONSECUTIVE BYTE LOCATIONS WHICH ARE HITS IN GROUP 0.
                                        ;*THE REMAINING FOUR PATTERNS ARE WRITTEN IN CONSECUTIVE
                                        ;*BYTE LOCATIONS WHICH ARE HITS IN GROUP 1. EACH PATTERN IS
                                        ;*READ BACK CHECKED AND THE COMPLIMENT PATTERN IS WRITTEN.
                                        ;*AFTER ALL THE PATTERNS HAVE BEEN CHECKED AND COMPLEMENTED
                                        ;*THE COMPLIMENTED PATTERNS ARE CHECKED.
                                        ;*
                                        ;;*********************************************************************
033676  000004                          TST34:  SCOPE
033700  012737  000040  001676                  MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                                                        ;SET THE SKAD REGISTER
033706  012737  035540  054230                  MOV     #TST35,SKAD     ;IN CASE THE TEST ABORTS.

033714  012737  054076  000114                  MOV     #SPUR,@#CACHVEC ;ADDRESS AND PUT THE NO ERROR
033722  113737  001502  001626                  MOVB    $TSTNM,$TMP0    ;EXPECTED ROUTINES ADDRESS IN
                                                                        ;THE PARITY ERROR VECTOR.

033730  012737  001001  035376          MM1:    MOV     #001001,MMPAT1  ;SET UP THE PATTERN
033736  012737  004004  035400                  MOV     #004004,MMPAT2  ;REGISTERS.
033744  012737  020020  035402                  MOV     #020020,MMPAT3
033752  012737  100100  035404                  MOV     #100100,MMPAT4

033760  012700  035410                          MOV     #MMTMP2,R0      ;LOCATE THE TEST AREA IN
033764  042700  176003                          BIC     #176003,R0      ;MEMORY WHOSE 'HITS' WILL NOT
033770  010001                                  MOV     R0,R1           ;INTERFER WITH HITS CAUSED
033772  062701  140000                          ADD     #TESTR1,R1      ;BY EXECUTING THIS TEST'S
033776  010002                                  MOV     R0,R2           ;CODE.
034000  062702  142000                          ADD     #TESTR2,R2

034004  010137  001640                          MOV     R1,$TMP5        ;SAVE THE TEST AREA ADDRESSES
034010  010137  001642                          MOV     R1,$TMP6                ;FOR ERROR PRINT OUT.
034014  062737  000002  001642                  ADD     #2,$TMP6
034022  010237  001644                          MOV     R2,$TMP7
034026  010237  001646                          MOV     R2,$TMP10
034032  062737  000002  001646                  ADD     #2,$TMP10

034040  012737  034046  001512                  MOV     #MM2,$LPERR     ;SET THE LOOP ON ERROR REGISTER.

034046  000240                          MM2:    NOP
034050  012737  000034  177746                  MOV     #S0M0M1,@#CONTRL        ;MAKE THE TEST AREAS HITS
034056  005711                                  TST     (R1)            ;IN GROUP 0 AND 1.
034060  005761  000002                          TST     2(R1)
034064  012737  000054  177746                  MOV     #S1M0M1,@#CONTRL
034072  005712                                  TST     (R2)
034074  005762  000002                          TST     2(R2)
034100  005037  177746                          CLR     @#CONTRL


034104  005711                                  TST     (R1)

                                                                        ;SEE IF REFERENCE ADDRESS
034106  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
034114  001006                                  BNE     MM3
                                                                        ;IF NOT ERROR!
```

```
034116  010137  001632                  MOV     R1,$TMP2
034122  012737  000000  001630          MOV     #0,$TMP1
034130  104001                          ERROR   1



034132                          MM3:

034132  005761  000002                  TST     2(R1)

                                                                ;SEE IF REFERENCE ADDRESS
034136  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
034144  001011                          BNE     MM4
                                                                ;IF NOT ERROR!
034146  010137  001632                  MOV     R1,$TMP2
034152  062737  000002  001632          ADD     #2,$TMP2
034160  012737  000000  001630          MOV     #0,$TMP1
034166  104001                          ERROR   1



034170                          MM4:

034170  005712                          TST     (R2)

                                                                ;SEE IF REFERENCE ADDRESS
034172  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
034200  001006                          BNE     MM5
                                                                ;IF NOT ERROR!
034202  010237  001632                  MOV     R2,$TMP2
034206  012737  000001  001630          MOV     #1,$TMP1
034214  104001                          ERROR   1



034216                          MM5:

034216  005762  000002                  TST     2(R2)

                                                                ;SEE IF REFERENCE ADDRESS
034222  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
034230  001014                          BNE     MM6
                                                                ;IF NOT ERROR!
034232  010237  001632                  MOV     R2,$TMP2
034236  062737  000002  001632          ADD     #2,$TMP2
034244  012737  000001  001630          MOV     #1,$TMP1
034252  104001                          ERROR   1



034254  012737  034262  001512          MOV     #MM6,$LPERR     ;SET LOOP ON ERROR ADDRESS
034262  012703  000001          MM6:    MOV     #1,R3
034266  012704  000004                  MOV     #4,R4
034272  110321                  MM7:    MOVB    R3,(R1)+        ;PUT THE TEST PATTERN
034274  006103                          ROL     R3              ;IN GROUP 0
034276  077403                          SOB     R4,MM7
```

```

034300  012704  000004                     MOV     #4,R4
034304  110322                     MM8:    MOVB    R3,(R2)+            ;PUT THE TEST PATTERN
034306  006103                             ROL     R3                  ;IN GROUP 1
034310  077403                             SOB     R4,MM8
034312  010001                             MOV     R0,R1
034314  062701  140000                     ADD     #TESTR1,R1          ;RE-ESTABLISH POINTERS TO
034320  010002                             MOV     R0,R2               ;THE TEST LOCATIONS.
034322  062702  142000                     ADD     #TESTR2,R2
034326  012703  035376                     MOV     #MMPAT1,R3          ;PUT THE ADDRESS OF THE TEST
                                                                       ;PATTERN REGISTERS IN R3

034332  005005                             CLR     R5


034334  005005                             CLR     R5
034336  111105                             MOVB    (R1),R5             ;GET THE PATTERN OUT OF
034340  032737  000010  177752             BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034346  001006                             BNE     MM9                 ;IS A HIT
034350  010137  001632                     MOV     R1,$TMP2
034354  012737  000000  001630             MOV     #0,$TMP1
034362  104001                             ERROR   1

034364  042705  177400             MM9:    BIC     #177400,R5
034370  022705  000001                     CMP     #1,R5   ;SEE IF THE DATA IS CORRECT.
034374  001402                             BEQ     MM10
034376  004737  035420                     JSR     PC,MMERR1
034402  105121                     MM10:   COMB    (R1)+               ;COMPLIMENT THE TEST PATTERN
034404  012713  001376                     MOV     #001376,(R3)



034410  005005                             CLR     R5
034412  111105                             MOVB    (R1),R5             ;GET THE PATTERN OUT OF
034414  032737  000010  177752             BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034422  001006                             BNE     MM11                ;IS A HIT
034424  010137  001632                     MOV     R1,$TMP2
034430  012737  000000  001630             MOV     #0,$TMP1
034436  104001                             ERROR   1

034440  042705  177400             MM11:   BIC     #177400,R5
034444  022705  000002                     CMP     #2,R5   ;SEE IF THE DATA IS CORRECT.
034450  001402                             BEQ     MM12
034452  004737  035420                     JSR     PC,MMERR1
034456  105121                     MM12:   COMB    (R1)+               ;COMPLIMENT THE TEST PATTERN
034460  012713  176776                     MOV     #176776,(R3)


034464  062703  000002                     ADD     #2,R3               ;POINT TO THE NEXT ELEMENT
                                                                       ;IN THE TEST PATTERN TABLE.

034470  005005                             CLR     R5
034472  111105                             MOVB    (R1),R5             ;GET THE PATTERN OUT OF
034474  032737  000010  177752             BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034502  001006                             BNE     MM13                ;IS A HIT
034504  010137  001632                     MOV     R1,$TMP2
```

```
034510  012737  000000  001630                MOV     #0,$TMP1
034516  104001                                ERROR   1

034520  042705  177400                MM13:   BIC     #177400,R5
034524  022705  000004                        CMP     #4,R5   ;SEE IF THE DATA IS CORRECT.
034530  001402                                BEQ     MM14
034532  004737  035420                        JSR     PC,MMERR1
034536  105121                        MM14:   COMB    (R1)+               ;COMPLIMENT THE TEST PATTERN
034540  012713  004373                        MOV     #004373,(R3)



034544  005005                                CLR     R5
034546  111105                                MOVB    (R1),R5             ;GET THE PATTERN OUT OF
034550  032737  000010  177752                BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034556  001006                                BNE     MM15                ;IS A HIT
034560  010137  001632                        MOV     R1,$TMP2
034564  012737  000000  001630                MOV     #0,$TMP1
034572  104001                                ERROR   1

034574  042705  177400                MM15:   BIC     #177400,R5
034600  022705  000010                        CMP     #10,R5  ;SEE IF THE DATA IS CORRECT.
034604  001402                                BEQ     MM16
034606  004737  035420                        JSR     PC,MMERR1
034612  105121                        MM16:   COMB    (R1)+               ;COMPLIMENT THE TEST PATTERN
034614  012713  173773                        MOV     #173773,(R3)


034620  062703  000002                        ADD     #2,R3               ;POINT TO THE NEXT ELEMENT
                                                                          ;IN THE TEST PATTERN TABLE.

034624  005005                                CLR     R5
034626  111205                                MOVB    (R2),R5             ;GET THE PATTERN OUT OF
034630  032737  000010  177752                BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034636  001006                                BNE     MM17                ;IS A HIT
034640  010237  001632                        MOV     R2,$TMP2
034644  012737  000001  001630                MOV     #1,$TMP1
034652  104001                                ERROR   1

034654  042705  177400                MM17:   BIC     #177400,R5
034660  022705  000020                        CMP     #20,R5  ;SEE IF THE DATA IS CORRECT.
034664  001402                                BEQ     MM18
034666  004737  035432                        JSR     PC,MMERR2
034672  105122                        MM18:   COMB    (R2)+               ;COMPLIMENT THE TEST PATTERN
034674  012713  020357                        MOV     #020357,(R3)



034700  005005                                CLR     R5
034702  111205                                MOVB    (R2),R5             ;GET THE PATTERN OUT OF
034704  032737  000010  177752                BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034712  001006                                BNE     MM19                ;IS A HIT
034714  010237  001632                        MOV     R2,$TMP2
034720  012737  000001  001630                MOV     #1,$TMP1
034726  104001                                ERROR   1

```

```
034730  042705  177400           MM19:   BIC     #177400,R5
034734  022705  000040                   CMP     #40,R5 ;SEE IF THE DATA IS CORRECT.
034740  001402                           BEQ     MM20
034742  004737  035432                   JSR     PC,MMERR2
034746  105122                   MM20:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
034750  012713  157757                   MOV     #157757,(R3)


034754  062703  000002                   ADD     #2,R3            ;POINT TO THE LAST ELEMENT
                                                                  ;IN THE TEST PATTERN TABLE.

034760  005005                           CLR     R5
034762  111205                           MOVB    (R2),R5          ;GET THE PATTERN OUT OF
034764  032737  000010  177752           BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
034772  001006                           BNE     MM21             ;IS A HIT
034774  010237  001632                   MOV     R2,$TMP2
035000  012737  000001  001630           MOV     #1,$TMP1
035006  104001                           ERROR   1

035010  042705  177400           MM21:   BIC     #177400,R5
035014  022705  000100                   CMP     #100,R5 ;SEE IF THE DATA IS CORRECT.
035020  001402                           BEQ     MM22
035022  004737  035432                   JSR     PC,MMERR2
035026  105122                   MM22:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
035030  012713  100277                   MOV     #100277,(R3)



035034  005005                           CLR     R5
035036  111205                           MOVB    (R2),R5          ;GET THE PATTERN OUT OF
035040  032737  000010  177752           BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
035046  001006                           BNE     MM23             ;IS A HIT
035050  010237  001632                   MOV     R2,$TMP2
035054  012737  000001  001630           MOV     #1,$TMP1
035062  104001                           ERROR   1

035064  042705  177400           MM23:   BIC     #177400,R5
035070  022705  000200                   CMP     #200,R5 ;SEE IF THE DATA IS CORRECT.
035074  001402                           BEQ     MM24
035076  004737  035432                   JSR     PC,MMERR2
035102  105122                   MM24:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
035104  012713  077677                   MOV     #077677,(R3)


035110  010001                           MOV     R0,R1            ;RE-ESTABLISH POINTERS TO
035112  062701  140000                   ADD     #TESTR1,R1       ;THE TEST AREA
035116  010002                           MOV     R0,R2
035120  062702  142000                   ADD     #TESTR2,R2


035124  012105                           MOV     (R1)+,R5         ;CHECK THE COMPLIMENTED

035126  005761  177776                   TST     -2(R1)


                                                                  ;SEE IF REFERENCE ADDRESS
035132  032737  000010  177752           BIT     #10,@#HITMIS     ;IS A HIT.
```

```
035140  001011                                   BNE     MM25
                                                                         ;IF NOT ERROR!
035142  010137  001632                           MOV     R1,$TMP2
035146  062737  177776  001632                   ADD     #-2,$TMP2
035154  012737  000000  001630                   MOV     #0,$TMP1
035162  104001                                   ERROR   1




035164  020537  035376                  MM25:    CMP     R5,MMPAT1                 ;IS PATTERN CORRECT?
035170  001402                                   BEQ     MM26
035172  004737  035462                           JSR     PC,MMERR4


035176                                  MM26:

035176  012105                                   MOV     (R1)+,R5        ;CHECK THE COMPLIMENTED

035200  005761  177776                           TST     -2(R1)

                                                                         ;SEE IF REFERENCE ADDRESS
035204  032737  000010  177752                   BIT     #10,@#HITMIS    ;IS A HIT.
035212  001011                                   BNE     MM27
                                                                         ;IF NOT ERROR!
035214  010137  001632                           MOV     R1,$TMP2
035220  062737  177776  001632                   ADD     #-2,$TMP2
035226  012737  000000  001630                   MOV     #0,$TMP1
035234  104001                                   ERROR   1




035236  020537  035400                  MM27:    CMP     R5,MMPAT2                 ;IS PATTERN CORRECT?
035242  001402                                   BEQ     MM28
035244  004737  035462                           JSR     PC,MMERR4


035250                                  MM28:

035250  012205                                   MOV     (R2)+,R5        ;CHECK THE COMPLIMENTED

035252  005762  177776                           TST     -2(R2)

                                                                         ;SEE IF REFERENCE ADDRESS
035256  032737  000010  177752                   BIT     #10,@#HITMIS    ;IS A HIT.
035264  001011                                   BNE     MM29
                                                                         ;IF NOT ERROR!
035266  010237  001632                           MOV     R2,$TMP2
035272  062737  177776  001632                   ADD     #-2,$TMP2
035300  012737  000001  001630                   MOV     #1,$TMP1
035306  104001                                   ERROR   1
```

```
035310  020537  035402                  MM29:     CMP     R5,MMPAT3                  ;IS PATTERN CORRECT?
035314  001402                                    BEQ     MM30
035316  004737  035502                            JSR     PC,MMERR5


035322                                  MM30:

035322  012205                                    MOV     (R2)+,R5              ;CHECK THE COMPLIMENTED

035324  005762  177776                            TST     -2(R2)

                                                                                ;SEE IF REFERENCE ADDRESS
035330  032737  000010  177752                    BIT     #10,@#HITMIS          ;IS A HIT.
035336  001011                                    BNE     MM31
                                                                                ;IF NOT ERROR!
035340  010237  001632                            MOV     R2,$TMP2
035344  062737  177776  001632                    ADD     #-2,$TMP2
035352  012737  000001  001630                    MOV     #1,$TMP1
035360  104001                                    ERROR   1




035362  020537  035404                  MM31:     CMP     R5,MMPAT4                  ;IS PATTERN CORRECT?
035366  001464                                    BEQ     MM32
035370  004737  035502                            JSR     PC,MMERR5

035374  000461                                    BR      MM32                  ;FINISHED THIS TEST.

035376  000000                          MMPAT1:   .WORD   0                     ;THIS IS THE TEST PATTERN
035400  000000                          MMPAT2:   .WORD   0                     ;TABLE.
035402  000000                          MMPAT3:   .WORD   0
035404  000000                          MMPAT4:   .WORD   0

035406  000000                          MMTMP1:   .WORD   0                     ;THIS AREA IS USED TO ESTABLISH
035410  000004                          MMTMP2:   .BLKW   4                     ;A TEST LOCATION WHOSE HITS WON'T
                                                                                ;BE INTERFERRED WITH BY THE CODE
                                                                                ;IN THE REST OF THIS TEST.

035420  005037  001630                  MMERR1:   CLR     $TMP1                 ;COME HERE TO REPORT
035424  010137  001636                            MOV     R1,$TMP4              ;GROUP 0 ERROR,WHILE READING
035430  000405                                    BR      MMERR3                ;A BYTE INTO R5

035432  012737  000001  001630          MMERR2:   MOV     #1,$TMP1                    ;COME HERE TO REPORT
035440  010237  001636                            MOV     R2,$TMP4              ;GROUP 1 ERROR, READING A
                                                                                ;BYTE INTO R5.
035444  012637  001632                  MMERR3:   MOV     (SP)+,$TMP2
035450  010537  001634                            MOV     R5,$TMP3

035454  104017                                    ERROR   17
035456  000177  144150                            JMP     @$TMP2

035462  005037  001630                  MMERR4:   CLR     $TMP1                 ;REPORT AN ERROR IN GROUP
035466  010137  001636                            MOV     R1,$TMP4              ;0 WHILE READING A WORD
035472  062737  177776  001636                    ADD     #-2,$TMP4
035500  000410                                    BR      MMERR6
```

```

035502  012737  000001  001630  MMERR5: MOV     #1,$TMP1
035510  010237  001636                  MOV     R2,$TMP4
035514  062737  177776  001636          ADD     #-2,$TMP4

035522  012637  001632          MMERR6: MOV     (SP)+,$TMP2
035526  010537  001634                  MOV     R5,$TMP3

035532  104020                          ERROR   20
035534  000177  144072                  JMP     @$TMP2

035540                          MM32:                           ;DONE!




                                ;;*****************************************************************
                                ;*TEST 35        CACHE ARBITRATION AND HIGH SPEED I/O TEST
                                ;*
                                ;*THIS IS A TEST OF:
                                ;*       1.      CACHE ARBITRATION
                                ;*       2.      THE MASS BUS AND UNIBUS PORTS TO THE CACHE
                                ;*       3.      HIGH SPEED I/O THROUGH THE CACHE
                                ;*
                                ;*IT MAKE USE OF THE FOLLOWING DEVICES:
                                ;*       RS04
                                ;*       RP04
                                ;*       RK05
                                ;*       MASS BUSS TESTER
                                ;*       UNIBUS EXERCISER
                                ;*
                                ;*IF ANY OF THESE DEVICES ARE PRESENT AND WRITE ENABLED THE WILL BE USED
                                ;*IN THIS TEST. ONLY THE LOWEST WRITE ENABLED DRIVE NUMBER OF EACH DEVICE
                                ;*WILL BE USED.
                                ;*
                                ;*       CAUTION!!!
                                ;*       THIS TEST WILL WRITE ON THE DISKS IT USES. SO VITAL SYSTEMS
                                ;*       DISKS SHOULD BE REMOVED OR WRITE PROTECTED BEFORE RUNNING
                                ;*       THIS DIAGNOSTIC.
                                ;*
                                ;*IF UNIT ZERO OF A PARTICULAR DEVICE IS WRITE PROTECTED THEN THIS TEST
                                ;*WILL TRY TO USE UNIT ONE, ETC.
                                ;*
                                ;*ALL AVAILABLE DEVICES ARE STARTED DOING TRANSFERS AT THE SAME TIME
                                ;*TO DIFFERENT PARTS OF MEMORY.
                                ;*EACH DEVICE HAS A CONTROL ROUTINE WHICH DRIVES THAT DEVICE THROUGH
                                ;*THE CYCLE:
                                ;*       1.      WRITE A RANDOM DATA PATTERN IN MEMORY
                                ;*       2.      COPY THAT PATTERN ONTO THE DISK
                                ;*       3.      WRITE CHECK THE DISK
                                ;*       4.      READ THE PATTERN OFF THE DISK BACK INTO MEMORY
                                ;*       5.      CHECK DATA
                                ;*       6.      START OVER AT 1.
                                ;*
                                ;*EACH DEVICE IS CAUSED TO GO THROUGH THIS CYCLE A PREDETERMINED
                                ;*NUMBER OF TIMES. THIS NUMBER IS CONTAINED IN THE LOCATION,
                                        ;*CYCNT, AND CAN BE CHANGED BY THE USER AT THE CONSOLE TO ANY VALUE
                                        ;*HE DESIRES).
                                        ;*
                                        ;*INTERRUPTS ARE ENABLED SO THAT IT IS POSSIBLE TO GET MANY DEVICES
                                        ;*DOING TRANSFERS AT ONCE.
                                        ;*
                                        ;*UNFORTUNATELY THE DEGREE TO WHICH FAULTS CAN BE ISOLATED IS
                                        ;*LIMITED BY THE FACT THAT THERE ARE MANY ELEMENTS, DEVICES, INVOLVED.
                                        ;*THESE ERRORS ARE REPORTED:
                                        ;*                 1.      ALL DEVICE ERRORS
                                        ;*                 2.      ALL DATA OR PARITY ERRORS
                                        ;*
                                        ;*NOTE THAT THIS NOT INTENDED TO BE USED AS AN I/O DEVICE DIAGNOSTIC!
                                        ;*ALL THE DEVICES WHICH ARE USED ARE ASSUMED TO BE IN PROPER WORKING
                                        ;*CONDITION.
                                        ;*
                                        ;*
                                        ;;*********************************************************************
035540  000004                          TST35:  SCOPE

                                                                          ;SET THE SKAD REGISTER
035542  012737  042206  054230                  MOV     #TST36,SKAD       ;IN CASE THE TEST ABORTS.

035550  104416                                  RSET
035552  113737  001502  001626                  MOVB    $TSTNM,$TMP0

035560  012700  172340                          MOV     #KIPAR0,R0        ;INITIALLY PUT MEMORY
035564  012701  077406                          MOV     #77406,R1         ;MANAGEMENT IN A 'PASSIVE'
035570  012702  172300                          MOV     #KIPDR0,R2        ;STATE, THAT IS MAP ALL
035574  012703  000010                          MOV     #10,R3            ;VIRTUAL ADDRESSES ON TO
035600  010122                          64$:    MOV     R1,(R2)+          ;THEMSELVES AS PHYSICAL
035602  077302                                  SOB     R3,64$            ;ADDRESSES.
035604  005020                                  CLR     (R0)+
035606  012720  000200                          MOV     #200,(R0)+
035612  012720  000400                          MOV     #400,(R0)+
035616  012720  000600                          MOV     #600,(R0)+
035622  012720  001000                          MOV     #1000,(R0)+
035626  012720  001200                          MOV     #1200,(R0)+
035632  012720  001400                          MOV     #1400,(R0)+
035636  012710  177600                          MOV     #177600,(R0)

035642  012737  000001  177572                  MOV     #1,@#MMR0
035650  012737  000060  172516                  MOV     #60,@#MMR3

035656  004737  041752                  INT0:   JSR     PC,GTBINT                         ;INITIALIZE THE MEMORY BUFFER
                                                                                          ;ALLOCATION ROUTINES.
035662  004737  055456                          JSR     PC,SIZDEV                         ;GO DETERMINE WHAT DEVICES ARE
                                                                                          ;PRESENT.
035666  005046                                  CLR     -(SP)                             ;MAKE THE WAIT LOOP ACCESSABLE
035670  012746  036140                          MOV     #WAITLP,-(SP)                     ;TO AN 'RTI'.

035674  012700  056050                  INT1:   MOV     #RS4DFL,R0                        ;GET READY TO SEE WHAT DEVICES
035700  012701  036070                          MOV     #RS4CR,R1                         ;ARE TO BE USED.
035704  012702  036102                          MOV     #RS4SUN,R2
035710  012703  036114                          MOV     #RS4ASS,R3
035714  012704  000005                          MOV     #5,R4

```

```
035720  005011                  INT2:   CLR     (R1)                    ;CLEAR THE UNIT NUMBER.
035722  005012                          CLR     (R2)                    ;CLEAR THE COUNTER.
035724  105710                          TSTB    (R0)                    ;IS THERE A DRIVE.
035726  001447                          BEQ     INT6                    ;BRANCH IF NOT.

035730  111005                          MOVB    (R0),R5                 ;OTHERWISE DETERMINE A UNIT NUM.
035732  104412                          SAVREG
035734  012700  000010                  MOV     #10,R0
035740  005001                          CLR     R1
035742  012702  000001                  MOV     #1,R2
035746  030205                  INT3:   BIT     R2,R5
035750  001405                          BEQ     INT4
035752  010137  036064                  MOV     R1,INTMP1
035756  104414                          RESREG
035760  000137  036000                  JMP     INT5
035764  005201                  INT4:   INC     R1
035766  006302                          ASL     R2
035770  077012                          SOB     R0,INT3
035772  104414                          RESREG
035774  000137  036046                  JMP     INT6

036000  013711  036066          INT5:   MOV     CYCNT,(R1)              ;FOUND THE DRIVE SO SET UP THE
036004  020127  036070                  CMP     R1,#RS4CR
036010  001001                          BNE     1$
036012  006311                          ASL     (R1)
036014  020127  036072          1$:     CMP     R1,#RP4CR
036020  001001                          BNE     2$
036022  006311                          ASL     (R1)
036024  020127  036074          2$:     CMP     R1,#RH4CR
036030  001001                          BNE     3$
036032  006311                          ASL     (R1)
036034  012746  000340          3$:     MOV     #340,-(SP)              ;PASS COUNT AND MAKE THE DRIVER
036040  011346                          MOV     (R3),-(SP)              ;ACCESSIBLE BY A 'RTI'.
036042  013712  036064                  MOV     INTMP1,(R2)

036046  005200                  INT6:   INC     R0
036050  005721                          TST     (R1)+           ;MOVE THE POINTERS TO THE NEXT DEVICE.
036052  022223                          CMP     (R2)+,(R3)+
036054  000240                          NOP
036056  077460                          SOB     R4,INT2


036060  000240                          NOP
036062  000002                          RTI             -               ;START THE TEST!


                                ;THESE ARE SOME TABLES THAT ARE USED TO CONTROL AND SET UP THIS TEST.
036064  000000                  INTMP1: .WORD   0


036066  000010                  CYCNT:  .WORD   10                      ;THE PASS COUNT!!!!

036070  000000                  RS4CR:  .WORD   0                       ;PASS COUNT FOR EACH DEVICE.
036072  000000                  RP4CR:  .WORD   0
036074  000000                  RH4CR:  .WORD   0
036076  000000                  RK5CR:  .WORD   0
```

```
036100  000000                          UBECR:  .WORD   0

036102  000000                          RS4SUN: .WORD   0                               ;THE DRIVE NUMBER USED FOR EACH
036104  000000                          RP4SUN: .WORD   0                               ;DEVICE.
036106  000000                          RH4SUN: .WORD   0
036110  000000                          RK5SUN: .WORD   0
036112  000000                          UBESUN: .WORD   0

        036114                          SETBLE=RS4ASS
036114  036204                          RS4ASS: .WORD   DRRS4                           ;STARTING ADDRESSES OF EACH DRIVER.
036116  037016                          RP4ASS: .WORD   DRRP4
036120  037630                          RH4ASS: .WORD   DRRH4
036122  040422                          RK5ASS: .WORD   DRRK5
036124  041234                          UBEASS: .WORD   DRUBE

036126  000000                          RS4RB:  .WORD   0                               ;WRITE AND READ BUFFERS OF EACH DEVICE.
036130  000000                          RP4RB:  .WORD   0
036132  000000                          RH4RB:  .WORD   0
036134  000000                          RK5RB:  .WORD   0
036136  000000                          UBERB:  .WORD   0


                                        ;THIS IS THE WAIT ROUTINE. COME HERE WHEN WAITING FOR AN INTERRUPT
                                        ;OR WHEN DONE, ALL THE PASS COUNTS HAVE GONE TO ZERO.
036140  000230                          WAITLP: SPL     0               ;LOWER THE PRIORITY.
036142  005737  036076                          TST     RK5CR           ;WAIT FOR INTERRUPT OR ZERO PASS COUNT.
036146  001374                                  BNE     WAITLP
036150  005737  036100                          TST     UBECR
036154  001371                                  BNE     WAITLP
036156  005737  036072                          TST     RP4CR
036162  001366                                  BNE     WAITLP
036164  005737  036070                          TST     RS4CR
036170  001363                                  BNE     WAITLP
036172  005737  036074                          TST     RH4CR
036176  001360                                  BNE     WAITLP

036200  000137  042204                          JMP     INDONE          ;FINISHED!!!



                                        ;THIS IS THE RS4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                        ;TEST.

036204  000240                          DRRS4:  NOP
036206  012737  007007  037012                  MOV     #7007,DRS4T1    ;INITIALIZE THE RANDOM DISK ADDRESS
036214  012737  006006  037014                  MOV     #6006,DRS4T2    ;GENERATER.
036222  012737  005005  036466                  MOV     #5005,RS4AA3

036230  000240                          RS4AA:  NOP
036232  000240                                  NOP
036234  104412                                  SAVREG
036236  004737  042066                          JSR     PC,GETBUF       ;GET A MEMORY BUFFER.
036242  036126                                  .WORD   RS4RB
036244  013701  036126                          MOV     RS4RB,R1
036250  005000                                  CLR     R0
036252  073027  000014                          ASHC    #12.,R0
```

```
 6814
 6815  036256  000237                              SPL     7                        ;GET A RANDOM DISK ADDRESS.
 6816  036260  013737  037012  053106              MOV     DRS4T1,$HINUM
 6817  036266  013737  037014  053110              MOV     DRS4T2,$LONUM
 6818  036274  004737  053010                      JSR     PC,$RAND
 6819  036300  013737  053106  037012              MOV     $HINUM,DRS4T1
 6820  036306  013737  053110  037014              MOV     $LONUM,DRS4T2
 6821  036314  000230                              SPL     0
 6822
 6823  036316  013702  036102                      MOV     RS4SUN,R2                          ;SET UP THE DEVICE UNIT NUM.
 6824  036322  110237  036663                      MOVB    R2,RS4112
 6825  036326  110237  036511                      MOVB    R2,RS4BB
 6826  036332  110237  036555                      MOVB    R2,RS4HH
 6827  036336  110237  036621                      MOVB    R2,RS4NN
 6828
 6829  036342  013703  037012                      MOV     DRS4T1,R3                          ;SET UP THE DISK ADDRESS.
 6830  036346  013704  037014                      MOV     DRS4T2,R4
 6831  036352  010337  036512                      MOV     R3,RS4CC
 6832  036356  010337  036664                      MOV     R3,RS4113
 6833  036362  010337  036556                      MOV     R3,RS4II
 6834  036366  010337  036622                      MOV     R3,RS4OO
 6835  036372  010437  036514                      MOV     R4,RS4DD
 6836  036376  010437  036560                      MOV     R4,RS4JJ
 6837  036402  010437  036666                      MOV     R4,RS4114
 6838  036406  010437  036624                      MOV     R4,RS4PP
 6839
 6840  036412  010137  036470                      MOV     R1,RS4AA1                          ;SET THE MEMORY ADDRESS.
 6841  036416  010137  036516                      MOV     R1,RS4EE
 6842  036422  010137  036562                      MOV     R1,RS4KK
 6843  036426  010137  036626                      MOV     R1,RS4QQ
 6844  036432  010137  036670                      MOV     R1,RS4115
 6845  036436  010037  036630                      MOV     R0,RS4RR
 6846  036442  010037  036672                      MOV     R0,RS4116
 6847  036446  010037  036472                      MOV     R0,RS4AA2
 6848  036452  010037  036520                      MOV     R0,RS4FF
 6849  036456  010037  036564                      MOV     R0,RS4LL
 6850
 6851  036462  104414                              RESREG
 6852
 6853  036464  104440                              WRRAND                           ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
 6854  036466  000000                    RS4AA3:   .WORD   0
 6855  036470  000000                    RS4AA1:   .WORD   0
 6856  036472  000000                    RS4AA2:   .WORD   0
 6857  036474  004000                              .WORD   4000
 6858  036476  005237  036466                      INC     RS4AA3
 6859
 6860  036502  000240                              NOP
 6861  036504  000237                              SPL     7
 6862  036506  104442                              CALRS4            ;GET THE RS4 TO DO THE TRANSFER FROM MEMORY
 6863  036510     161                              .BYTE   161
 6864  036511     000                    RS4BB:    .BYTE   0
 6865  036512  000000                    RS4CC:    .WORD   0
 6866  036514  000000                    RS4DD:    .WORD   0
 6867  036516  000000                    RS4EE:    .WORD   0
 6868  036520  000000                    RS4FF:    .WORD   0
 6869  036522  004000                              .WORD   4000
```

```
036524  036542                           .WORD   RS4GG

036526  000240                           NOP
036530  004737  036740                   JSR     PC,RS4YY
036534  005066  000002                   CLR     2(SP)
036540  000002                           RTI                     ;GO DO SOMETHING ELSE WHILE WAITING
                                                                 ;FOR THE INTERRUPT!

036542  000240                   RS4GG:  NOP
036544  004737  036740                   JSR     PC,RS4YY                 ;SEE IF THERE WERE ANY ERRORS.

036550  000237                           SPL     7
036552  104442                           CALRS4          ;DO THE WRITE CHECK
036554     151                           .BYTE   151
036555     000                   RS4HH:  .BYTE   0
036556  000000                   RS4II:  .WORD   0
036560  000000                   RS4JJ:  .WORD   0
036562  000000                   RS4KK:  .WORD   0
036564  000000                   RS4LL:  .WORD   0
036566  004000                           .WORD   4000
036570  036606                           .WORD   RS4MM

036572  000240                           NOP
036574  004737  036740                   JSR     PC,RS4YY
036600  005066  000002                   CLR     2(SP)
036604  000002                           RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.

036606  000240                   RS4MM:  NOP
036610  004737  036740                   JSR     PC,RS4YY        ;SEE IF THERE WERE ANY ERRORS.


036614  000237                           SPL     7
036616  104442                           CALRS4          ;READ THE DISK.
036620     171                           .BYTE   171
036621     000                   RS4NN:  .BYTE   0
036622  000000                   RS4OO:  .WORD   0
036624  000000                   RS4PP:  .WORD   0
036626  000000                   RS4QQ:  .WORD   0
036630  000000                   RS4RR:  .WORD   0
036632  004000                           .WORD   4000
036634  036652                           .WORD   RS4111

036636  000240                           NOP
036640  004737  036740                   JSR     PC,RS4YY
036644  005066  000002                   CLR     2(SP)
036650  000002                           RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.

036652  004737  036740           RS4111: JSR     PC,RS4YY
036656  000237                           SPL     7

036660  104442                           CALRS4
036662     151                           .BYTE   151
036663     000                   RS4112: .BYTE   0
036664  000000                   RS4113: .WORD   0
036666  000000                   RS4114: .WORD   0
036670  000000                   RS4115: .WORD   0
```

```
036672  000000                          RS4116: .WORD   0
036674  004000                                  .WORD   4000
036676  036714                                  .WORD   RS4SS
036700  000240                                  NOP
036702  004737  036740                          JSR     PC,RS4YY
036706  005066  000002                          CLR     2(SP)
036712  000002                                  RTI

036714  000240                          RS4SS:  NOP
036716  004737  036740                          JSR     PC,RS4YY                ;SEE IF ANY ERRORS OCCURRED.

036722  005337  036070                          DEC     RS4CR                   ;DECRIMENT THE PASS COUNT.
036726  001001                                  BNE     RS4XX                   ;IF NOT DONE CONTINUE.
036730  000002                                  RTI                             ;IF DONE GET OUT!

036732  000240                          RS4XX:  NOP
036734  000137  036230                          JMP     RS4AA                   ;RESTART.

036740  000240                          RS4YY:  NOP
036742  005737  057374                          TST     RS4ER1  ;SEE IF ANY ERRORS OCCURRED.
036746  001420                                  BEQ     RS4ZZ                   ;IF NOT THEN RETURN TO CALL.

036750  000237                                  SPL     7
036752  005037  036070                          CLR     RS4CR                   ;IF YES THEN CLEAR THE PASS COUNT.
036756  013737  057376  001630                  MOV     RS4ER2,$TMP1            ;AND MAKE AN ERROR CALL.
036764  013737  057402  001634                  MOV     RS4ER4,$TMP3
036772  013737  057400  001632                  MOV     RS4ER3,$TMP2
037000  104154                                  ERROR   154
037002  000230                                  SPL     0
037004  005726                                  TST     (SP)+
037006  000002                                  RTI                             ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                                                ;FROM THE TEST.

037010  000207                          RS4ZZ:  RTS     PC                      ;THERE WERE NO ERRORS.

037012  000000                          DRS4T1: .WORD   0
037014  000000                          DRS4T2: .WORD   0



                                        ;THIS IS THE RP4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                        ;TEST.

037016  000240                          DRRP4:  NOP
037020  012737  004004  037624                  MOV     #4004,DRP4T1            ;INITIALIZE THE RANDOM DISK ADDRESS
037026  012737  003003  037626                  MOV     #3003,DRP4T2            ;GENERATER.
037034  012737  002002  037300                  MOV     #2002,RP4AA3

037042  000240                          RP4AA:  NOP
037044  000240                                  NOP
037046  104412                                  SAVREG
037050  004737  042066                          JSR     PC,GETBUF               ;GET A MEMORY BUFFER.
037054  036130                                  .WORD   RP4RB
037056  013701  036130                          MOV     RP4RB,R1
037062  005000                                  CLR     R0
037064  073027  000014                          ASHC    #12.,R0
```

```

037070  000237                              SPL     7                     ;GET A RANDOM DISK ADDRESS.
037072  013737  037624  053106              MOV     DRP4T1,$HINUM
037100  013737  037626  053110              MOV     DRP4T2,$LONUM
037106  004737  053010                      JSR     PC,$RAND
037112  013737  053106  037624              MOV     $HINUM,DRP4T1
037120  013737  053110  037626              MOV     $LONUM,DRP4T2
037126  000230                              SPL     0

037130  013702  036104                      MOV     RP4SUN,R2                     ;SET UP THE DEVICE UNIT NUM.
037134  110237  037475                      MOVB    R2,RP4112
037140  110237  037323                      MOVB    R2,RP4BB
037144  110237  037367                      MOVB    R2,RP4HH
037150  110237  037433                      MOVB    R2,RP4NN

037154  013703  037624                      MOV     DRP4T1,R3                     ;SET UP THE DISK ADDRESS.
037160  013704  037626                      MOV     DRP4T2,R4
037164  010337  037324                      MOV     R3,RP4CC
037170  010337  037476                      MOV     R3,RP4113
037174  010337  037370                      MOV     R3,RP4II
037200  010337  037434                      MOV     R3,RP4OO
037204  010437  037326                      MOV     R4,RP4DD
037210  010437  037372                      MOV     R4,RP4JJ
037214  010437  037500                      MOV     R4,RP4114
037220  010437  037436                      MOV     R4,RP4PP

037224  010137  037302                      MOV     R1,RP4AA1                     ;SET THE MEMORY ADDRESS.
037230  010137  037330                      MOV     R1,RP4EE
037234  010137  037374                      MOV     R1,RP4KK
037240  010137  037440                      MOV     R1,RP4QQ
037244  010137  037502                      MOV     R1,RP4115
037250  010037  037442                      MOV     R0,RP4RR
037254  010037  037504                      MOV     R0,RP4116
037260  010037  037304                      MOV     R0,RP4AA2
037264  010037  037332                      MOV     R0,RP4FF
037270  010037  037376                      MOV     R0,RP4LL

037274  104414                              RESREG

037276  104440                              WRRAND                ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
037300  000000                     RP4AA3:  .WORD   0
037302  000000                     RP4AA1:  .WORD   0
037304  000000                     RP4AA2:  .WORD   0
037306  004000                              .WORD   4000
037310  005237  037300                      INC     RP4AA3

037314  000240                              NOP
037316  000237                              SPL     7
037320  104444                              CALRP4          ;GET THE RP4 TO DO THE TRANSFER FROM MEMORY
037322     161                              .BYTE   161
037323     000                     RP4BB:   .BYTE   0
037324  000000                     RP4CC:   .WORD   0
037326  000000                     RP4DD:   .WORD   0
037330  000000                     RP4EE:   .WORD   0
037332  000000                     RP4FF:   .WORD   0
037334  004000                              .WORD   4000
```

```
 7038  037336  037354                          .WORD   RP4GG
 7039
 7040  037340  000240                          NOP
 7041  037342  004737  037552                  JSR     PC,RP4YY
 7042  037346  005066  000002                  CLR     2(SP)
 7043  037352  000002                          RTI                             ;GO DO SOMETHING ELSE WHILE WAITING
 7044                                                                          ;FOR THE INTERRUPT!
 7045
 7046  037354  000240                 RP4GG:   NOP
 7047  037356  004737  037552                  JSR     PC,RP4YY                          ;SEE IF THERE WERE ANY ERRORS.
 7048
 7049  037362  000237                          SPL     7
 7050  037364  104444                          CALRP4             ;DO THE WRITE CHECK
 7051  037366     151                          .BYTE   151
 7052  037367     000                 RP4HH:   .BYTE   0
 7053  037370  000000                 RP4II:   .WORD   0
 7054  037372  000000                 RP4JJ:   .WORD   0
 7055  037374  000000                 RP4KK:   .WORD   0
 7056  037376  000000                 RP4LL:   .WORD   0
 7057  037400  004000                          .WORD   4000
 7058  037402  037420                          .WORD   RP4MM
 7059
 7060  037404  000240                          NOP
 7061  037406  004737  037552                  JSR     PC,RP4YY
 7062  037412  005066  000002                  CLR     2(SP)
 7063  037416  000002                          RTI                             ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.
 7064
 7065  037420  000240                 RP4MM:   NOP
 7066  037422  004737  037552                  JSR     PC,RP4YY                ;SEE IF THERE WERE ANY ERRORS.
 7067
 7068
 7069  037426  000237                          SPL     7
 7070  037430  104444                          CALRP4             ;READ THE DISK.
 7071  037432     171                          .BYTE   171
 7072  037433     000                 RP4NN:   .BYTE   0
 7073  037434  000000                 RP4OO:   .WORD   0
 7074  037436  000000                 RP4PP:   .WORD   0
 7075  037440  000000                 RP4QQ:   .WORD   0
 7076  037442  000000                 RP4RR:   .WORD   0
 7077  037444  004000                          .WORD   4000
 7078  037446  037464                          .WORD   RP4111
 7079
 7080  037450  000240                          NOP
 7081  037452  004737  037552                  JSR     PC,RP4YY
 7082  037456  005066  000002                  CLR     2(SP)
 7083  037462  000002                          RTI                ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.
 7084
 7085  037464  004737  037552         RP4111:  JSR     PC,RP4YY
 7086  037470  000237                          SPL     7
 7087
 7088  037472  104444                          CALRP4
 7089  037474     151                          .BYTE   151
 7090  037475     000                 RP4112:  .BYTE   0
 7091  037476  000000                 RP4113:  .WORD   0
 7092  037500  000000                 RP4114:  .WORD   0
 7093  037502  000000                 RP4115:  .WORD   0
```

```
037504  000000                  RP4116: .WORD   0
037506  004000                          .WORD   4000
037510  037526                          .WORD   RP4SS
037512  000240                          NOP
037514  004737  037552                  JSR     PC,RP4YY
037520  005066  000002                  CLR     2(SP)
037524  000002                          RTI

037526  000240                  RP4SS:  NOP
037530  004737  037552                  JSR     PC,RP4YY                ;SEE IF ANY ERRORS OCCURRED.

037534  005337  036072                  DEC     RP4CR                   ;DECRIMENT THE PASS COUNT.
037540  001001                          BNE     RP4XX                   ;IF NOT DONE CONTINUE.
037542  000002                          RTI                             ;IF DONE GET OUT!

037544  000240                  RP4XX:  NOP
037546  000137  037042                  JMP     RP4AA                   ;RESTART.

037552  000240                  RP4YY:  NOP
037554  005737  056424                  TST     RP4ER1  ;SEE IF ANY ERRORS OCCURRED.
037560  001420                          BEQ     RP4ZZ                   ;IF NOT THEN RETURN TO CALL.

037562  000237                          SPL     7
037564  005037  036072                  CLR     RP4CR                   ;IF YES THEN CLEAR THE PASS COUNT.
037570  013737  056426  001630          MOV     RP4ER2,$TMP1            ;AND MAKE AN ERROR CALL.
037576  013737  056432  001634          MOV     RP4ER4,$TMP3
037604  013737  056430  001632          MOV     RP4ER3,$TMP2
037612  104155                          ERROR   155
037614  000230                          SPL     0
037616  005726                          TST     (SP)+
037620  000002                          RTI                             ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                                        ;FROM THE TEST.

037622  000207                  RP4ZZ:  RTS     PC                      ;THERE WERE NO ERRORS.

037624  000000                  DRP4T1: .WORD   0
037626  000000                  DRP4T2: .WORD   0



                                ;THIS IS THE RH4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                ;TEST.

037630  000240                  DRRH4:  NOP
037632  012737  070070  040416          MOV     #70070,DRH4T1           ;INITIALIZE THE RANDOM DISK ADDRESS
037640  012737  060060  040420          MOV     #60060,DRH4T2           ;GENERATER.
037646  012737  050050  040072          MOV     #50050,RH4AA3

037654  000240                  RH4AA:  NOP
037656  000240                          NOP
037660  104412                          SAVREG
037662  004737  042066                  JSR     PC,GETBUF               ;GET A MEMORY BUFFER.
037666  036132                          .WORD   RH4RB
037670  013701  036132                  MOV     RH4RB,R1
037674  005000                          CLR     R0
037676  073027  000014                  ASHC    #12.,R0
```

```

037702  000237                              SPL     7                              ;GET A RANDOM DISK ADDRESS.
037704  013737  040416  053106              MOV     DRH4T1,$HINUM
037712  013737  040420  053110              MOV     DRH4T2,$LONUM
037720  004737  053010                      JSR     PC,$RAND
037724  013737  053106  040416              MOV     $HINUM,DRH4T1
037732  013737  053110  040420              MOV     $LONUM,DRH4T2
037740  000230                              SPL     0

037742  013702  036106                      MOV     RH4SUN,R2                            ;SET UP THE DEVICE UNIT NUM.
037746  110237  040267                      MOVB    R2,RH4112
037752  110237  040115                      MOVB    R2,RH4BB
037756  110237  040161                      MOVB    R2,RH4HH
037762  110237  040225                      MOVB    R2,RH4NN

037766  013703  040416                      MOV     DRH4T1,R3                            ;SET UP THE DISK ADDRESS.
037772  013704  040420                      MOV     DRH4T2,R4
037776  010337  040116                      MOV     R3,RH4CC
040002  010337  040270                      MOV     R3,RH4113
040006  010337  040162                      MOV     R3,RH4II
040012  010337  040226                      MOV     R3,RH4OO

040016  010137  040074                      MOV     R1,RH4AA1                            ;SET THE MEMORY ADDRESS.
040022  010137  040122                      MOV     R1,RH4EE
040026  010137  040166                      MOV     R1,RH4KK
040032  010137  040232                      MOV     R1,RH4QQ
040036  010137  040274                      MOV     R1,RH4115
040042  010037  040234                      MOV     R0,RH4RR
040046  010037  040276                      MOV     R0,RH4116
040052  010037  040076                      MOV     R0,RH4AA2
040056  010037  040124                      MOV     R0,RH4FF
040062  010037  040170                      MOV     R0,RH4LL

040066  104414                              RESREG

040070  104440                              WRRAND                         ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
040072  000000                     RH4AA3:  .WORD   0
040074  000000                     RH4AA1:  .WORD   0
040076  000000                     RH4AA2:  .WORD   0
040100  004000                              .WORD   4000
040102  005237  040072                      INC     RH4AA3

040106  000240                              NOP
040110  000237                              SPL     7
040112  104446                              CALRH4                ;GET THE RH4 TO DO THE TRANSFER FROM MEMORY
040114     161                              .BYTE   161
040115     000                     RH4BB:   .BYTE   0
040116  000000                     RH4CC:   .WORD   0
040120  000000                     RH4DD:   .WORD   0
040122  000000                     RH4EE:   .WORD   0
040124  000000                     RH4FF:   .WORD   0
040126  004000                              .WORD   4000
040130  040146                              .WORD   RH4GG

040132  000240                              NOP
040134  004737  040344                      JSR     PC,RH4YY
```

```
040140  005066  000002                    CLR     2(SP)
040144  000002                            RTI                             ;GO DO SOMETHING ELSE WHILE WAITING
                                                                          ;FOR THE INTERRUPT!

040146  000240                  RH4GG:    NOP
040150  004737  040344                    JSR     PC,RH4YY                          ;SEE IF THERE WERE ANY ERRORS.

040154  000237                            SPL     7
040156  104446                            CALRH4                ;DO THE WRITE CHECK
040160     171                            .BYTE   171
040161     000                  RH4HH:    .BYTE   0
040162  000000                  RH4II:    .WORD   0
040164  000000                  RH4JJ:    .WORD   0
040166  000000                  RH4KK:    .WORD   0
040170  000000                  RH4LL:    .WORD   0
040172  004000                            .WORD   4000
040174  040212                            .WORD   RH4MM

040176  000240                            NOP
040200  004737  040344                    JSR     PC,RH4YY
040204  005066  000002                    CLR     2(SP)
040210  000002                            RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.

040212  000240                  RH4MM:    NOP
040214  004737  040344                    JSR     PC,RH4YY        ;SEE IF THERE WERE ANY ERRORS.


040220  000237                            SPL     7
040222  104446                            CALRH4                ;READ THE DISK.
040224     151                            .BYTE   151
040225     000                  RH4NN:    .BYTE   0
040226  000000                  RH4OO:    .WORD   0
040230  000000                  RH4PP:    .WORD   0
040232  000000                  RH4QQ:    .WORD   0
040234  000000                  RH4RR:    .WORD   0
040236  004000                            .WORD   4000
040240  040256                            .WORD   RH4111

040242  000240                            NOP
040244  004737  040344                    JSR     PC,RH4YY
040250  005066  000002                    CLR     2(SP)
040254  000002                            RTI                   ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.

040256  004737  040344          RH4111:   JSR     PC,RH4YY
040262  000237                            SPL     7

040264  104446                            CALRH4
040266     171                            .BYTE   171
040267     000                  RH4112:   .BYTE   0
040270  000000                  RH4113:   .WORD   0
040272  000000                  RH4114:   .WORD   0
040274  000000                  RH4115:   .WORD   0
040276  000000                  RH4116:   .WORD   0
040300  004000                            .WORD   4000
040302  040320                            .WORD   RH4SS
040304  000240                            NOP
```

```
040306  004737  040344                      JSR     PC,RH4YY
040312  005066  000002                      CLR     2(SP)
040316  000002                              RTI

040320  000240                    RH4SS:    NOP
040322  004737  040344                      JSR     PC,RH4YY            ;SEE IF ANY ERRORS OCCURRED.

040326  005337  036074                      DEC     RH4CR               ;DECRIMENT THE PASS COUNT.
040332  001001                              BNE     RH4XX               ;IF NOT DONE CONTINUE.
040334  000002                              RTI                         ;IF DONE GET OUT!

040336  000240                    RH4XX:    NOP
040340  000137  037654                      JMP     RH4AA               ;RESTART.

040344  000240                    RH4YY:    NOP
040346  005737  062076                      TST     RH4ER1  ;SEE IF ANY ERRORS OCCURRED.
040352  001420                              BEQ     RH4ZZ               ;IF NOT THEN RETURN TO CALL.

040354  000237                              SPL     7
040356  005037  036074                      CLR     RH4CR               ;IF YES THEN CLEAR THE PASS COUNT.
040362  013737  062100  001630              MOV     RH4ER2,$TMP1        ;AND MAKE AN ERROR CALL.
040370  013737  062104  001634              MOV     RH4ER4,$TMP3
040376  013737  062102  001632              MOV     RH4ER3,$TMP2
040404  104156                              ERROR   156
040406  000230                              SPL     0
040410  005726                              TST     (SP)+
040412  000002                              RTI                         ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                                        ;FROM THE TEST.

040414  000207                    RH4ZZ:    RTS     PC                  ;THERE WERE NO ERRORS.

040416  000000                    DRH4T1:   .WORD   0
040420  000000                    DRH4T2:   .WORD   0




                                  ;THIS IS THE RK5 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                  ;TEST.

040422  000240                    DRRK5:    NOP
040424  012737  030030  041230              MOV     #30030,DRK5T1       ;INITIALIZE THE RANDOM DISK ADDRESS
040432  012737  040040  041232              MOV     #40040,DRK5T2       ;GENERATER.
040440  012737  050050  040704              MOV     #50050,RK5AA3

040446  000240                    RK5AA:    NOP
040450  000240                              NOP
040452  104412                              SAVREG
040454  004737  042066                      JSR     PC,GETBUF           ;GET A MEMORY BUFFER.
040460  036134                              .WORD   RK5RB
040462  013701  036134                      MOV     RK5RB,R1
040466  005000                              CLR     R0
040470  073027  000014                      ASHC    #12.,R0

040474  000237                              SPL     7                   ;GET A RANDOM DISK ADDRESS.
040476  013737  041230  053106              MOV     DRK5T1,$HINUM
```

```
 040504  013737  041232  053110                 MOV     DRK5T2,$LONUM
 040512  004737  053010                         JSR     PC,$RAND
 040516  013737  053106  041230                 MOV     $HINUM,DRK5T1
 040524  013737  053110  041232                 MOV     $LONUM,DRK5T2
 040532  000230                                 SPL     0

 040534  013702  036110                         MOV     RK5SUN,R2                       ;SET UP THE DEVICE UNIT NUM.
 040540  110237  041101                         MOVB    R2,RK5112
 040544  110237  040727                         MOVB    R2,RK5BB
 040550  110237  040773                         MOVB    R2,RK5HH
 040554  110237  041037                         MOVB    R2,RK5NN

 040560  013703  041230                         MOV     DRK5T1,R3                       ;SET UP THE DISK ADDRESS.
 040564  013704  041232                         MOV     DRK5T2,R4
 040570  010337  040730                         MOV     R3,RK5CC
 040574  010337  041102                         MOV     R3,RK5113
 040600  010337  040774                         MOV     R3,RK5II
 040604  010337  041040                         MOV     R3,RK5OO
 040610  010437  040732                         MOV     R4,RK5DD
 040614  010437  040776                         MOV     R4,RK5JJ
 040620  010437  041104                         MOV     R4,RK5114
 040624  010437  041042                         MOV     R4,RK5PP

 040630  010137  040706                         MOV     R1,RK5AA1                       ;SET THE MEMORY ADDRESS.
 040634  010137  040734                         MOV     R1,RK5EE
 040640  010137  041000                         MOV     R1,RK5KK
 040644  010137  041044                         MOV     R1,RK5QQ
 040650  010137  041106                         MOV     R1,RK5115
 040654  010037  041046                         MOV     R0,RK5RR
 040660  010037  041110                         MOV     R0,RK5116
 040664  010037  040710                         MOV     R0,RK5AA2
 040670  010037  040736                         MOV     R0,RK5FF
 040674  010037  041002                         MOV     R0,RK5LL

 040700  104414                                 RESREG

 040702  104440                                 WRRAND                  ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
 040704  000000                       RK5AA3:   .WORD   0
 040706  000000                       RK5AA1:   .WORD   0
 040710  000000                       RK5AA2:   .WORD   0
 040712  004000                                 .WORD   4000
 040714  005237  040704                         INC     RK5AA3

 040720  000240                                 NOP
 040722  000237                                 SPL     7
 040724  104450                                 CALRK5          ;GET THE RK5 TO DO THE TRANSFER FROM MEMORY
 040726     103                                 .BYTE   103
 040727     000                       RK5BB:    .BYTE   0
 040730  000000                       RK5CC:    .WORD   0
 040732  000000                       RK5DD:    .WORD   0
 040734  000000                       RK5EE:    .WORD   0
 040736  000000                       RK5FF:    .WORD   0
 040740  004000                                 .WORD   4000
 040742  040760                                 .WORD   RK5GG

 040744  000240                                 NOP
```

```
040746  004737  041156            JSR     PC,RK5YY
040752  005066  000002            CLR     2(SP)
040756  000002                    RTI                                    ;GO DO SOMETHING ELSE WHILE WAITING
                                                                         ;FOR THE INTERRUPT!

040760  000240          RK5GG:    NOP
040762  004737  041156            JSR     PC,RK5YY                                 ;SEE IF THERE WERE ANY ERRORS.

040766  000237                    SPL     7
040770  104450                    CALRK5                 ;DO THE WRITE CHECK
040772     107                    .BYTE   107
040773     000          RK5HH:    .BYTE   0
040774  000000          RK5II:    .WORD   0
040776  000000          RK5JJ:    .WORD   0
041000  000000          RK5KK:    .WORD   0
041002  000000          RK5LL:    .WORD   0
041004  004000                    .WORD   4000
041006  041024                    .WORD   RK5MM

041010  000240                    NOP
041012  004737  041156            JSR     PC,RK5YY
041016  005066  000002            CLR     2(SP)
041022  000002                    RTI                                    ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.

041024  000240          RK5MM:    NOP
041026  004737  041156            JSR     PC,RK5YY                       ;SEE IF THERE WERE ANY ERRORS.


041032  000237                    SPL     7
041034  104450                    CALRK5                 ;READ THE DISK.
041036     105                    .BYTE   105
041037     000          RK5NN:    .BYTE   0
041040  000000          RK5OO:    .WORD   0
041042  000000          RK5PP:    .WORD   0
041044  000000          RK5QQ:    .WORD   0
041046  000000          RK5RR:    .WORD   0
041050  004000                    .WORD   4000
041052  041070                    .WORD   RK5111

041054  000240                    NOP
041056  004737  041156            JSR     PC,RK5YY
041062  005066  000002            CLR     2(SP)
041066  000002                    RTI                    ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.

041070  004737  041156  RK5111:   JSR     PC,RK5YY
041074  000237                    SPL     7

041076  104450                    CALRK5
041100     107                    .BYTE   107
041101     000          RK5112:   .BYTE   0
041102  000000          RK5113:   .WORD   0
041104  000000          RK5114:   .WORD   0
041106  000000          RK5115:   .WORD   0
041110  000000          RK5116:   .WORD   0
041112  004000                    .WORD   4000
041114  041132                    .WORD   RK5SS
```

```
041116  000240                            NOP
041120  004737  041156                    JSR     PC,RK5YY
041124  005066  000002                    CLR     2(SP)
041130  000002                            RTI

041132  000240                   RK5SS:   NOP
041134  004737  041156                    JSR     PC,RK5YY          ;SEE IF ANY ERRORS OCCURRED.

041140  005337  036076                    DEC     RK5CR             ;DECRIMENT THE PASS COUNT.
041144  001001                            BNE     RK5XX             ;IF NOT DONE CONTINUE.
041146  000002                            RTI                       ;IF DONE GET OUT!

041150  000240                   RK5XX:   NOP
041152  000137  040446                    JMP     RK5AA             ;RESTART.

041156  000240                   RK5YY:   NOP
041160  005737  060330                    TST     RK5ER1  ;SEE IF ANY ERRORS OCCURRED.
041164  001420                            BEQ     RK5ZZ             ;IF NOT THEN RETURN TO CALL.

041166  000237                            SPL     7
041170  005037  036076                    CLR     RK5CR             ;IF YES THEN CLEAR THE PASS COUNT.
041174  013737  060332  001630            MOV     RK5ER2,$TMP1      ;AND MAKE AN ERROR CALL.
041202  013737  060336  001634            MOV     RK5ER4,$TMP3
041210  013737  060334  001632            MOV     RK5ER3,$TMP2
041216  104160                            ERROR   160
041220  000230                            SPL     0
041222  005726                            TST     (SP)+
041224  000002                            RTI                       ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                                    ;FROM THE TEST.

041226  000207                   RK5ZZ:   RTS     PC                ;THERE WERE NO ERRORS.

041230  000000                   DRK5T1:  .WORD   0
041232  000000                   DRK5T2:  .WORD   0



                                 ;THIS IS THE UBE DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                 ;TEST.
041234  012737  050050  041602   DRUBE:   MOV     #50050,DUBET1     ;INITIALIZE THE RANDOM DATA
041242  012737  060060  041604            MOV     #60060,DUBET2     ;GENERATER.
041250  012737  070070  041410            MOV     #70070,UBEAA3

041256  104412                   UBEAA:   SAVREG
041260  004737  042066                    JSR     PC,GETBUF         ;PICK UP A MEMORY BUFFER
041264  036136                            .WORD   UBERB

041266  013701  036136                    MOV     UBERB,R1          ;COMPUTE THE MEMORY ADDRESS.
041272  005000                            CLR     R0
041274  073027  000014                    ASHC    #12.,R0
041300  010137  041412                    MOV     R1,UBEAA1
041304  010137  041436                    MOV     R1,UBEDD
041310  010137  041476                    MOV     R1,UBEII
041314  010037  041414                    MOV     R0,UBEAA2
041320  010037  041440                    MOV     R0,UBEEE
041324  010037  041500                    MOV     R0,UBEJJ
```

```

041330  000237                                  SPL     7
041332  013737  041602  053106                  MOV     DUBET1,$HINUM
041340  013737  041604  053110                  MOV     DUBET2,$LONUM
041346  004737  053010                          JSR     PC,$RAND
041352  013737  053106  041602                  MOV     $HINUM,DUBET1
041360  013737  053110  041604                  MOV     $LONUM,DUBET2
041366  000230                                  SPL     0

041370  013703  041602                          MOV     DUBET1,R3          ;SET THE UNIBUS TESTER DATA REG.
041374  010337  041474                          MOV     R3,UBEHH
041400  010337  041434                          MOV     R3,UBECCC

041404  104414                                  RESREG

041406  104440                                  WRRAND                     ;FILL THE MEMORY BUFFER WITH
041410  000000                          UBEAA3: .WORD   0                  ;RANDOM DATA.
041412  000000                          UBEAA1: .WORD   0
041414  000000                          UBEAA2: .WORD   0
041416  004000                                  .WORD   4000
041420  005237  041410                          INC     UBEAA3

041424  000237                                  SPL     7
041426  104452                                  CALUBE                     ;DO A READ MEMORY FUNCTION.
041430  042543                                  .WORD   42543
041432  000000                          UBEBB:  .WORD   0
041434  000000                          UBECCC: .WORD   0
041436  000000                          UBEDD:  .WORD   0
041440  000000                          UBEEE:  .WORD   0
041442  010000                                  .WORD   10000
```

```
041444  041460                           .WORD   UBEFF

041446  004737  041540                   JSR     PC,UBEYY
041452  005066  000002                   CLR     2(SP)
041456  000002                           RTI                     ;GO DO SOMETHING ELSE WHILE
                                                                 ;WAITING FOR INTERRUPT.
041460  004737  041540          UBEFF:   JSR     PC,UBEYY

041464  000237                           SPL     7
041466  104452                           CALUBE                  ;DO A WRITE MEMORY FUNCTION.
041470  042543                           .WORD   42543
041472  000000                  UBEGG:   .WORD   0
041474  000000                  UBEHH:   .WORD   0
041476  000000                  UBEII:   .WORD   0
041500  000000                  UBEJJ:   .WORD   0
041502  010000                           .WORD   10000
041504  041520                           .WORD   UBEKK

041506  004737  041540                   JSR     PC,UBEYY
041512  005066  000002                   CLR     2(SP)
041516  000002                           RTI                     ;GO DO SOMETHING ELSE WHILE
                                                                 ;WAITING FOR THE INTERRUPT.
041520  004737  041540          UBEKK:   JSR     PC,UBEYY

041524  005337  036100                   DEC     UBECR           ;DECREMENT THE PASS COUNT.
041530  001001                           BNE     UBELL           ;BR IF NOT DONE

041532  000002                           RTI                     ;IF DONE RETURN.
041534  000137  041256          UBELL:   JMP     UBEAA           ;IF NOT DONE DO ANOTHER PASS.

041540  005737  061344          UBEYY:   TST     UBEER1          ;WERE THERE ANY ERRORS?
041544  001415                           BEQ     UBEZZ           ;BR IF NO.

041546  000237                           SPL     7               ;IF THERE WERE REPORT DEVICE FAILURE.
041550  005037  036100                   CLR     UBECR
041554  013737  061346  001630           MOV     UBEER2,$TMP1
041562  013737  061350  001632           MOV     UBEER3,$TMP2
041570  104161                           ERROR   161
041572  005726                           TST     (SP)+
041574  000230                           SPL     0
041576  000002                           RTI                     ;RETURN WITH THIS DRIVER LOCKED OUT.
041600  000207                  UBEZZ:   RTS     PC              ;NO ERRORS CONTINUE.

041602  000000                  DUBET1:  .WORD   0
041604  000000                  DUBET2:  .WORD   0



                                ;THIS ROUTINE IS USED TO GENERATE A BUFFER FULL OF RANDOM DATA.
                                ;IT IS CALLED USING THE TRAP TABLE CALL:
                                ;        WRRAND
                                ;        .WORD   HIGHNUM
                                ;        .WORD   LOADRS
                                ;        .WORD   HIGHADRS
                                ;        .WORD   WORDCOUNT
                                ;RET:
                                        ;WHERE HIGHNUM IS THE HIGH ORDER PART OF THE NUMBER USED TP PRIME THE
                                        ;RANDOM NUMBER GENERATER. THE LOW ORDER PART OF THAT NUMBER IS ASSUMED
                                        ;TO BE ZERO. LOADRS AND HIGHADRS IS THE 22 BIT ADDRESS OF THE BUFFER
                                        ;IN MEMORY WHICH WILL BE FILLED. WORDCOUNT IS THE NUMBER OF LOCATIONS
                                        ;TO BE WRITTEN.
041606  000237                          RANDWR: SPL     7
041610  011637  041750                          MOV     (SP),RANDTP
041614  062716  000010                          ADD     #10,(SP)
041620  104412                                  SAVREG
041622  013700  041750                          MOV     RANDTP,R0
041626  012001                                  MOV     (R0)+,R1
041630  012002                                  MOV     (R0)+,R2
041632  012003                                  MOV     (R0)+,R3
041634  012004                                  MOV     (R0)+,R4
041636  010237  041746                          MOV     R2,RLWT
041642  010337  041744                          MOV     R3,RHWT
041646  010137  053106                          MOV     R1,$HINUM
041652  005037  053110                          CLR     $LONUM

041656  013702  041744                  1$:     MOV     RHWT,R2         ;COMPUTE THE VIRTUAL ADDRESS OF THE BUFFER WORD.
041662  013703  041746                          MOV     RLWT,R3
041666  073227  177772                          ASHC    #-6,R2
041672  010337  172354                          MOV     R3,@#KIPAR6
041676  013702  041746                          MOV     RLWT,R2
041702  042702  177700                          BIC     #177700,R2
041706  062702  140000                          ADD     #140000,R2
041712  004737  053010                          JSR     PC,$RAND
041716  013712  053106                          MOV     $HINUM,(R2)
041722  062737  000002  041746                  ADD     #2,RLWT
041730  005537  041744                          ADC     RHWT
041734  077430                                  SOB     R4,1$

041736  000230                                  SPL     0
041740  104414                                  RESREG
041742  000002                                  RTI

041744  000000                          RHWT:   .WORD   0
041746  000000                          RLWT:   .WORD   0
041750  000000                          RANDTP: .WORD   0

                                        ;THIS ROUTINE IS USED TO INITIALIZE THE GET BUFFER ROUTINE.
041752  012700  036126                  GTBINT: MOV     #RS4RB,R0       ;CLEAR ALL THE BUFFER POINTERS.
041756  012701  000005                          MOV     #5,R1

041762  005020                          1$:     CLR     (R0)+
041764  077102                                  SOB     R1,1$
041766  104424                                  SIZE                    ;COMPUTE THE SIZE OF MEMORY.
041770  000000                          GTBILO: .WORD   0
041772  000000                          GTBIHI: .WORD   0
041774  062737  000002  041770                  ADD     #2,GTBILO
042002  005537  041772                          ADC     GTBIHI
042006  013700  041772                          MOV     GTBIHI,R0       ;COMPUTE THE 2K BLOCK SIZE OF MEMORY.
042012  013701  041770                          MOV     GTBILO,R1
042016  073027  177764                          ASHC    #-12.,R0
042022  010137  042054                          MOV     R1,GTMSIZ
042026  162701  000011                          SUB     #11,R1
```

```
042032  010137  042056                  MOV     R1,AVMBL
042036  012737  123456  042060          MOV     #123456,GTRNL
042044  012737  123456  042062          MOV     #123456,GTRNH
042052  000207                          RTS     PC

042054  000000                  GTMSIZ: .WORD   0
042056  000000                  AVMBL:  .WORD   0
042060  000000                  GTRNL:  .WORD   0
042062  000000                  GTRNH:  .WORD   0
042064  000000                  GETMP1: .WORD   0

                                ;THIS ROUTINE IS CALLED TO ALLOCATE A MEMORY BUFFER OF 2K WORDS LENGTH.
                                ;IT IS CALLED USING A JSR PC INSTRUCTION FOLLOWED BY THE TABLE ENTRY
                                ;OF RS4RB TO BE UPDATED.
042066  000237                  GETBUF: SPL     7                       ;LOCK OUT INTERRUPTS.
042070  011637  042064                  MOV     (SP),GETMP1
042074  062716  000002                  ADD     #2,(SP)                 ;PICK UP A POINTER TO THE ARGUMENT
                                                                        ;AND UPDATE THE RETURN ADDRESS.
042100  104412                          SAVREG
042102  013737  042060  053110  1$:     MOV     GTRNL,$LONUM
042110  013737  042062  053106          MOV     GTRNH,$HINUM
042116  004737  053010                  JSR     PC,$RAND
042122  013737  053110  042060          MOV     $LONUM,GTRNL
042130  013701  053106                  MOV     $HINUM,R1
042134  010137  042062                  MOV     R1,GTRNH
042140  005000                          CLR     R0
042142  071037  042056                  DIV     AVMBL,R0

042146  012702  036126                  MOV     #RS4RB,R2               ;SEE IF THIS AREA IS ALREADY IN USE.
042152  012703  000005                  MOV     #5,R3
042156  062701  000011                  ADD     #11,R1

042162  020122                  2$:     CMP     R1,(R2)+
042164  001746                          BEQ     1$                      ;IF IT IS THEN TRY AGAIN.
042166  077303                          SOB     R3,2$

042170  017704  177670                  MOV     @GETMP1,R4              ;OTHERWISE GIVE THIS BUFFER TO THE DRIVER.
042174  010114                          MOV     R1,(R4)
042176  104414                          RESREG
042200  000230                          SPL     0
042202  000207                          RTS     PC


042204  104416                  INDONE: RSET





                                ;;*****************************************************************
                                ;*TEST 36        MASS BUS WRITE HIT CYCLE, INVALIDATION TEST
                                ;*
                                ;*THIS IS A TEST OF CACHE INVALIDATION ON MASS BUS CYCLES WHICH ARE
                                ;*WRITE HITS IN THE CACHE. A GROUP OF LOCATIONS IS MADE HITS AND THEN A
                                ;*MASS BUS DEVICE IS CALLED UPON TO DO TRANSFERS, WRITES TO THOSE
                                ;*LOCATIONS. THOSE WRITES SHOULD THUS BE INVALIDATED.
                                ;*
                                ;;*******************************************************************
042206  000004                  TST36:  SCOPE
                                                                ;SET THE SKAD REGISTER
042210  012737  047736  054230          MOV     #KT,SKAD        ;IN CASE THE TEST ABORTS.

042216  104416                          RSET
042220  113737  001502  001626          MOVB    $TSTNM,$TMP0
042226  004737  055456                  JSR     PC,SIZDEV       ;DETERMINE WHAT DEVICES ARE AVAILABLE.
042232  113737  056050  042746          MOVB    RS4DFL,RS4FT
042240  113737  056051  042747          MOVB    RP4DFL,RP4FT
042246  113737  056052  042750          MOVB    RH4DFL,RH4FT

042254  000137  043064          NN1:    JMP     NNDEV           ;GO COMPUTE THE DRIVE NUMBERS.

042260  005037  042744          NN2:    CLR     NNGRPF          ;FLAG WHICH DESIGNATES WHICH GOUP IS BEING
042264  012737  000044  042742          MOV     #S1M0,NNGRM     ;TESTED ON THIS PASS.
042272  012737  000030  042740          MOV     #S0M1,NNGRS     ;TEST GROUP ZERO FIRST.

042300  004737  042754          NN3:    JSR     PC,NNSTUP       ;GO MAKE THE TEST ADDRESSES HITS
042304  004777  000426                  JSR     PC,@NNUD                  ;USE THE FIRST DEVICE.


042310  012700  140000                  MOV     #TESTR1,R0
042314  012701  000400                  MOV     #256.,R1        ;MAKE SURE THOSE ADDRESSES ARE MISSES.

042320  005710                  1$:     TST     (R0)
042322  032737  000010  177752          BIT     #10,@#HITMIS
042330  001430                          BEQ     2$

042332  013737  042744  001630          MOV     NNGRPF,$TMP1    ;GOT A HIT REPORT FAILURE.
042340  010037  001632                  MOV     R0,$TMP2
042344  005037  001634                  CLR     $TMP3
042350  023727  042736  042552          CMP     NNUD,#NNRS4     ;WAS THE RS4 DOING THE TRANSFER?
042356  001003                          BNE     11$             ;BRANCH IF NOT.
042360  104151                          ERROR   151
042362  000137  042420                  JMP     NN5
042366  023727  042736  042644  11$:    CMP     NNUD,#NNRP4     ;WAS IT THE RP4?
042374  001003                          BNE     12$
042376  104152                          ERROR   152
042400  000137  042420                  JMP     NN5
042404  104153                  12$:    ERROR   153
042406  000137  042420                  JMP     NN5

042412  062700  000004          2$:     ADD     #4,R0
042416  077140                          SOB     R1,1$

042420  005237  042744          NN5:    INC     NNGRPF          ;TESTED BOTH GROUPS?
042424  022737  000002  042744          CMP     #2,NNGRPF
042432  001410                          BEQ     NN6             ;BRANCH IF YES.
042434  012737  000044  042740          MOV     #S1M0,NNGRS     ;IF NOT GO BACK AND TEST GROUP ONE.
042442  012737  000030  042742          MOV     #S0M1,NNGRM
042450  000137  042300                  JMP     NN3

042454  000137  043332          NN6:    JMP     NNDONE

042460  104446                  NNRH4:  CALRH4          ;THIS IS THE CALL TO READ THE MASS BUS TESTER.
```

```
042462     071                          .BYTE   71
042463     000                  NNRH4U: .BYTE   0
042464  052525                          .WORD   52525
042466  000000                          .WORD   0
042470  140000                          .WORD   TESTR1
042472  000000                          .WORD   0
042474  001000                          .WORD   512.
042476  042510                          .WORD   2$

042500  005737  062076          1$:     TST     RH4ER1          ;ANY DEVICE ERRORS?
042504  100401                          BMI     2$              ;BRANCH IF YES.
042506  000207                          RTS     PC              ;IF NOT RETURN.

042510  013737  062100  001630  2$:     MOV     RH4ER2,$TMP1    ;REPORT DEVICE ERROR.
042516  013737  062102  001632          MOV     RH4ER3,$TMP2
042524  013737  062104  001634          MOV     RH4ER4,$TMP3
042532  005726                          TST     (SP)+
042534  104156                          ERROR   156
042536  105037  056052                  CLRB    RH4DFL
042542  105037  042750                  CLRB    RH4FT
042546  000137  042254                  JMP     NN1

042552  104442                  NNRS4:  CALRS4                  ;THIS IS A CALL TO DO AN RS4 READ.
042554     071                          .BYTE   71
042555     000                  NNRS4U: .BYTE   0
042556  000000                          .WORD   0
042560  000000                          .WORD   0
042562  140000                          .WORD   TESTR1
042564  000000                          .WORD   0
042566  001000                          .WORD   512.
042570  042602                          .WORD   2$

042572  005737  057374          1$:     TST     RS4ER1          ;SEE IF THERE WERE DEVICE ERRORS.
042576  100401                          BMI     2$              ;BR IF YES.
042600  000207                          RTS     PC

042602  013737  057376  001630  2$:     MOV     RS4ER2,$TMP1
042610  013737  057400  001632          MOV     RS4ER3,$TMP2
042616  013737  057402  001634          MOV     RS4ER4.$TMP3
042624  005726                          TST     (SP)+
042626  104154                          ERROR   154
042630  105037  056050                  CLRB    RS4DFL
042634  105037  042746                  CLRB    RS4FT
042640  000137  042254                  JMP     NN1

042644  104444                  NNRP4:  CALRP4                  ;THIS IS A CALL TO DO AN RP4 READ.
042646     071                          .BYTE   71
042647     000                  NNRP4U: .BYTE   0
042650  000000                          .WORD   0
042652  000000                          .WORD   0
042654  140000                          .WORD   TESTR1
042656  000000                          .WORD   0
042660  001000                          .WORD   512.
042662  042674                          .WORD   2$

042664  005737  056424          1$:     TST     RP4ER1          ;WERE THERE ANY DEVICE ERRORS?
```

```
042670  100401                           BMI     2$
042672  000207                           RTS     PC

042674  013737  056426  001630  2$:      MOV     RP4ER2,$TMP1
042702  013737  056430  001632           MOV     RP4ER3,$TMP2
042710  013737  056432  001634           MOV     RP4ER4,$TMP3
042716  005726                           TST     (SP)+
042720  104155                           ERROR   155
042722  105037  056051                   CLRB    RP4DFL
042726  105037  042747                   CLRB    RP4FT
042732  000137  042254                   JMP     NN1

042736  000000                  NNUD:    .WORD   0

042740  000000                  NNGRS:   .WORD   0
042742  000000                  NNGRM:   .WORD   0
042744  000000                  NNGRPF:  .WORD   0

                                ;THIS ROUTINE IS CALLED TO MAKE THE ADDRESSES IN TESTR1
                                ;HITS PRIOR TO CALLING FOR THE MB DEVICE TO DO TRANSFERS.
042746     000                  RS4FT:   .BYTE   0
042747     000                  RP4FT:   .BYTE   0
042750     000                  RH4FT:   .BYTE   0
042751     000                  RK5FT:   .BYTE   0
042752     000                  UBEFT:   .BYTE   0
        042754                           .EVEN

042754  104412                  NNSTUP:  SAVREG
042756  012700  042754                   MOV     #NNSTUP,R0        ;MAKE THIS CODE HITS IN THE
042762  012701  001000                   MOV     #512.,R1          ;GROUP NOT BEING TESTED.
042766  012702  142000                   MOV     #TESTR2,R2

042772  013737  042742  177746  1$:      MOV     NNGRM,@#CONTRL
043000  005720                           TST     (R0)+
043002  013737  042740  177746           MOV     NNGRS,@#CONTRL
043010  005722                           TST     (R2)+
043012  077111                           SOB     R1,1$

043014  013700  042740          2$:      MOV     NNGRS,R0
043020  042700  000014                   BIC     #14,R0
043024  010037  177746                   MOV     R0,@#CONTRL
043030  012701  140000                   MOV     #TESTR1,R1
043034  012702  001000                   MOV     #512.,R2
043040  005721                  3$:      TST     (R1)+
043042  077202                           SOB     R2,3$
043044  013700  042742                   MOV     NNGRM,R0
043050  042700  000014                   BIC     #14,R0
043054  010037  177746                   MOV     R0,@#CONTRL
043060  104414                           RESREG
043062  000207                           RTS     PC


                                ;SEE WHAT DEVICE TO USE NEXT.
043064  000240                  NNDEV:   NOP
043066  000240                           NOP
043070  005037  042736                   CLR     NNUD
```

```
 7852 043074 113700  042746                         MOVB    RS4FT,R0                ;IS THERE AN RS4 DRIVE.
 7853 043100 001430                                 BEQ     NND2                    ;BR IS NOT
 7854
 7855 043102 000240                       NND0:     NOP
 7856 043104 012701  000001                         MOV     #1,R1                   ;FIND OUT WHAT DRIVE  NUMBER IT IS.
 7857 043110 012737  042552  042736                 MOV     #NNRS4,NNUD
 7858 043116 005002                                 CLR     R2
 7859 043120 012703  000010                         MOV     #10,R3
 7860 043124 000240                       1$:       NOP
 7861 043126 030100                                 BIT     R1,R0
 7862 043130 001406                                 BEQ     2$
 7863 043132 140137  042746                         BICB    R1,RS4FT                ;FOUND IT.
 7864 043136 110237  042555                         MOVB    R2,NNRS4U
 7865 043142 000137  042260                         JMP     NN2
 7866 043146 005202                       2$:       INC     R2
 7867 043150 006301                                 ASL     R1
 7868 043152 077314                                 SOB     R3,1$                   ;KEEP LOOKING.
 7869
 7870 043154 104000                                 ERROR   0
 7871 043156 105037  042746                         CLRB    RS4FT
 7872
 7873 043162 000240                       NND2:     NOP
 7874 043164 113700  042747                         MOVB    RP4FT,R0                ;IS THERE AN RP04 DRIVE.
 7875 043170 001426                                 BEQ     NND3                    ;BR IF NO
 7876 043172 012701  000001                         MOV     #1,R1
 7877 043176 012737  042644  042736                 MOV     #NNRP4,NNUD
 7878 043204 005002                                 CLR     R2
 7879 043206 012703  000010                         MOV     #10,R3
 7880 043212 030100                       1$:       BIT     R1,R0
 7881 043214 001406                                 BEQ     2$
 7882 043216 140137  042746                         BICB    R1,RS4FT
 7883 043222 110237  042647                         MOVB    R2,NNRP4U
 7884 043226 000137  042260                         JMP     NN2
 7885 043232 005202                       2$:       INC     R2
 7886 043234 006301                                 ASL     R1
 7887 043236 077313                                 SOB     R3,1$
 7888 043240 104000                                 ERROR   0
 7889 043242 105037  042747                         CLRB    RP4FT
 7890
 7891 043246 000240                       NND3:     NOP
 7892 043250 113700  042750                         MOVB    RH4FT,R0                ;IS THERE A MASS BUS TESTER.
 7893 043254 001426                                 BEQ     NNDONE
 7894 043256 012701  000001                         MOV     #1,R1
 7895 043262 012737  042460  042736                 MOV     #NNRH4,NNUD
 7896 043270 005002                                 CLR     R2
 7897 043272 012703  000010                         MOV     #10,R3
 7898 043276 030100                       1$:       BIT     R1,R0
 7899 043300 001406                                 BEQ     2$
 7900 043302 140137  042750                         BICB    R1,RH4FT
 7901 043306 110237  042463                         MOVB    R2,NNRH4U
 7902 043312 000137  042260                         JMP     NN2
 7903 043316 005202                       2$:       INC     R2
 7904 043320 006301                                 ASL     R1
 7905 043322 077313                                 SOB     R3,1$
 7906 043324 104000                                 ERROR   0
 7907 043326 105037  042750                         CLRB    RH4FT
```

```
043332  104416                          NNDONE: RSET



043334  105737  001714                          TSTB    KB11CM                          ;11/74          (KB11CM)?
043340  001005                                  BNE     1$                              ;BRANCH IF YES
043342  105737  001713                          TSTB    KB11EM                          ;KB11-EM?
043346  001002                                  BNE     1$                              ;BR IF YES
043350  000137  047736                          JMP     KT                              ;GO TO KT IF NO
043354                                  1$:                                             ;ENTER HERE IF KB11-E

                                        ;;*******************************************************************
                                        ;*TEST 37               CHECK IVSS, VSIU BITS
                                                ;THIS TEST CHECKS THAT THE IVSS AND VSIU BITS OF THE CACHE
                                                ;CONTROL REGISTER CAN BE SET AND CLEARED.  VCIP IS ALSO
                                                ;CHECKED.
                                                ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                        ;;*******************************************************************
043354  000004                          TST37:  SCOPE
043356  005037  177746                          CLR     @#CONTRL
043362  005737  177746                          TST     @#CONTRL
043366  001404                                  BEQ     1$
043370  013737  177746  001556                  MOV     @#CONTRL,$REGO
043376  104055                                  ERROR   55      ;CCR COULD NOT BE CLEARED

043400  012737  040000  177746          1$:     MOV     #IVSS,@#CONTRL
043406  022737  040000  177746                  CMP     #IVSS,@#CONTRL
043414  001404                                  BEQ     2$
043416  013737  177746  001556                  MOV     @#CONTRL,$REGO
043424  104056                                  ERROR   56      ;IVSS COULD NOT BE SET

043426  042737  040000  177746          2$:     BIC     #IVSS,@#CONTRL
043434  001404                                  BEQ     3$
043436  013737  177746  001556                  MOV     @#CONTRL,$REGO
043444  104057                                  ERROR   57      ;IVSS COULD NOT BE CLEARED

043446  012737  020000  177746          3$:     MOV     #VSIU,@#CONTRL
043454  032737  020000  177746                  BIT     #VSIU,@#CONTRL
043462  001004                                  BNE     4$
043464  013737  177746  001556                  MOV     @#CONTRL,$REGO
043472  104060                                  ERROR   60      ;VSIU COULD NOT BE SET

043474  012700  000050                  4$:     MOV     #50,R0  ;WAIT FOR VCIP TO CLEAR
043500  032737  010000  177746                  BIT     #VCIP,@#CONTRL
043506  001405                                  BEQ     5$
043510  077007                                  SOB     R0,4$
043512  013737  177746  001556                  MOV     @#CONTRL,$REGO
043520  104061                                  ERROR   61                  ;VCIP DID NOT CLEAR WITHIN SOME
                                                                           ;SOME TIME AFTER VSIU WAS SET
043522  042737  020000  177746          5$:     BIC     #VSIU,@#CONTRL
043530  032737  020000  177746                  BIT     #VSIU,@#CONTRL
043536  001404                                  BEQ     6$
043540  013737  177746  001556                  MOV     @#CONTRL,$REGO
043546  104062                                  ERROR   62                   ;VSIU COULD NOT BE CLEARED
043550  032737  010000  177746          6$:     BIT     #VCIP,@#CONTRL
043556  001374                                  BNE     6$
```

```
                                        ;;*************************************************************
                                        ;*TEST 40        CHECK VSIU BIT, WITH IVSS ALREADY SET
                                                ;THIS TEST CHECKS THAT THE "VALID STORE IN USE" (VISU)
                                                ;BIT CAN BE SET AND CLEARED WHEN THE IVSS IS
                                                ;ALREADY SET.
                                                ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                        ;;*************************************************************
043560  000004                          TST40:  SCOPE

043562  012737  040000  177746                  MOV     #IVSS,@#CONTRL
043570  032737  020000  177746                  BIT     #VSIU,@#CONTRL
043576  001404                                  BEQ     1$
043600  013737  177746  001556                  MOV     @#CONTRL,$REGO
043606  104062                                  ERROR   62                      ;VALID STORE IN USE, BIT 13,
                                                                                ;COULD NOT BE CLEARED IN CCR
043610  032737  010000  177746          1$:     BIT     #VCIP,@#CONTRL
043616  001374                                  BNE     1$
043620  052737  020000  177746                  BIS     #VSIU,@#CONTRL
043626  032737  020000  177746                  BIT     #VSIU,@#CONTRL
043634  001004                                  BNE     2$
043636  013737  177746  001556                  MOV     @#CONTRL,$REGO
043644  104060                                  ERROR   60                      ;VSIU (BIT 13) COULD NOT BE SET
                                                                                ;IN CCR (IVSS WAS ALREADY SET).
043646  042737  020000  177746          2$:     BIC     #VSIU,@#CONTRL
043654  032737  020000  177746                  BIT     #VSIU,@#CONTRL
043662  001404                                  BEQ     TST41                   ;;EXIT
043664  013737  177746  001556                  MOV     @#CONTRL,$REGO
043672  104062                                  ERROR   62                      ;VSIU COULD NOT BE CLEARED IN CCR
                                                                                ;IVSS WAS ALREADY SET.

                                        ;;*************************************************************
                                        ;*TEST 41        CHECK VCIP SETS WHEN CF IS SET
                                                ;THIS TEST CHECKS THAT THE VCIP SETS WHEN CACHE-FLUSH IS
                                                ;DONE AND IT CLEARS OUT WITHIN A CERTAIN TIME AFTER
                                                ;THE FLUSH OF VALID STORE IS OVER
                                                ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                        ;;*************************************************************
043674  000004                          TST41:  SCOPE
043676  012737  000400  177746                  MOV     #FCAC, @# CONTRL;FLUSH CACHE
043704  000240                                  NOP
043706  012700  000062                          MOV     #50.,R0
043712  032737  010000  177746                  BIT     #VCIP, @# CONTRL
043720  001004                                  BNE     1$
043722  013737  177746  001556                  MOV     @#CONTRL,$REGO
043730  104063                                  ERROR   63      ;VCIP DID NOT SET WHEN CACHE
                                                                                ;FLUSH WAS ISSUED
043732  032737  010000  177746          1$:     BIT     #VCIP, @#CONTRL ;WAIT FOR VCIP TO CLEAR
043740  001405                                  BEQ     2$
043742  077005                                  SOB     R0,1$
043744  013737  177746  001556                  MOV     @#CONTRL,$REGO
043752  104061                                  ERROR   61                      ;VCIP DID NOT CLEAR WITHIN A
                                                                                ;CERTAIN TIME AFTER CACHE FLUSH
                                                                                ;WAS DONE
043754                                  2$:

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

```
                                      ;*TEST 42       CHECK CACHE FLUSH & VALID STORE SWITCHING
                                              ;THIS TEST CHECKS THAT WHEN A CACHE FLUSH IS DONE
                                              ;BY SETTING CF IN CCR, THE VALID STORE IN USE
                                              ;(VSIU) SURTCHES. VALID STORE SWITCHING FROM STORE-A
                                              ;TO STORE-B AND VICE-VERSA IS CHECKED
                                               ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                      ;;*****************************************************************
043754  000004                        TST42:  SCOPE
043756  005037  177746                        CLR     @#CONTRL
043762  032737  010000  177746  1$:           BIT     #VCIP, @#CONTRL
043770  001374                                BNE     1$
043772  012737  000400  177746                MOV     #FCAC, @#CONTRL ;FLUSH CACHE
044000  032737  020000  177746                BIT     #VSIU, @#CONTRL
044006  001004                                BNE     2$
044010  013737  177746  001556                MOV     @#CONTRL,$REGO
044016  104064                                ERROR   64         ;VSIU DID NOT SWITCH FROM 0 TO 1
                                                                 ;WHEN CACHE FLUSH WAS SET
044020  032737  010000  177746  2$:           BIT     #VCIP, @#CONTRL
044026  001374                                BNE     2$
044030  012737  000400  177746                MOV     #FCAC, @#CONTRL
044036  032737  020000  177746                BIT     #VSIU, @#CONTRL
044044  001404                                BEQ     3$
044046  013737  177746  001556                MOV     @#CONTRL,$REGO
044054  104064                                ERROR   64         ;VSIU DID NOT SWITCH FROM 1 TO 0 WHEN
                                                                 ;FLUSH-CACHE WAS SET IN CCR
044056  032737  010000  177746  3$:           BIT     #VCIP, @#CONTRL
044064  001374                                BNE     3$



                                      ;;*****************************************************************
                                      ;*TEST 43       CHECK IVSS INHIBITS SWITCHING OF VALID STORE IN USE
                                              ;THIS TEST CHECKS THAT WHEN ''INHIBIT VALID STORE SWITCHING''
                                              ;(IVSS) IS SET AND FLUSH-CACHE BIT IS SET, THE
                                              ;VALID STORE IN USE DOES NOT SWITCH
                                               ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                      ;;*****************************************************************
044066  000004                        TST43:  SCOPE
044070  005037  177746                        CLR     @#CONTRL
044074  032737  010000  177746  1$:           BIT     #VCIP,@#CONTRL
044102  001374                                BNE     1$
044104  012737  040000  177746                MOV     #IVSS, @#CONTRL ;SET IVSS
044112  052737  000400  177746                BIS     #FCAC, @#CONTRL ;FLUSH CACHE
044120  032737  020000  177746                BIT     #VSIU, @#CONTRL
044126  001404                                BEQ     2$              ;CHECK VSIU DID NOT SWITH
044130  013737  177746  001556                MOV     @#CONTRL,$REGO
044136  104065                                ERROR   65              ;VSIU SWITCHED, WHEN IVSS
                                                                      ;WAS SET AND CACHE FLUSH
                                                                      ;WAS DONE, IT SHOULD NOT SWITCH
044140  032737  010000  177746  2$:           BIT     #VCIP, @#CONTRL
044146  001374                                BNE     2$
044150  052737  020000  177746                BIS     #VSIU, @#CONTRL
044156  032737  010000  177746  3$:           BIT     #VCIP, @#CONTRL
044164  001374                                BNE     3$
044166  052737  000400  177746                BIS     #FCAC, @#CONTRL
044174  032737  020000  177746                BIT     #VSIU, @#CONTRL ;CHECK VSIU DID NOT SWITCH
044202  001004                                BNE     4$
```

```
044204  013737  177746  001556          MOV     @#CONTRL,$REG0
044212  104065                          ERROR   65                      ;VSIU SWITCHED, WHEN IVSS
                                                                        ;WAS SET AND CACHE FLUSH WAS
                                                                        ;DONE; IT SHOULD NOT SWITCH
044214  032737  010000  177746  4$:     BIT     #VCIP, @#CONTRL
044222  001374                          BNE     4$


                                ;;*****************************************************************
                                ;*TEST 44        CHECK VALID STORES (A & B) FOR GROUP 0
                                        ;THIS TEST CHECKS THE TWO VALID STORES (A&B) FOR GROUP 0
                                        ;OF THE CACHE. WHEN A CACHE-FLUSH IS ISSUED, THE CACHE
                                        ;SHOULD BE INVALIDATED BY SWITCHING THE VALID STORE
                                        ;IN USE
                                        ;THE TEST-CODE IS MADE HIT IN GROUP 1 (WHICH IS NOT
                                        ;BEING TESTED). THE TEST DATA IS MADE HIT IN GROUP 0.
                                        ;FLUSH-CACHE BIT IS SET IN THE CCR. IT IS CHECKED THAT
                                        ;THE TEST-DATA WHICH WAS HIT (MADE PREVIOUSLY) IN
                                        ;GROUP 0 IS NO MORE A HIT. EACH LOCATION OF THE
                                        ;TEST-DATA BLOCK IS REFERENCED AND CHECKED IF
                                        ;IT WAS A MISS. OTHERWISE AN ERROR IS REPORTED. AS A
                                        ;RESULT OF THE CACHE FLUSH THE VALID STORE SHOULD
                                        ;HAVE SWITCHED FROM 0 TO 1. THEN THE VALID STORE
                                        ;IS FORCED TO BE 0 AND THE TEST-DATA IS REFERENCED
                                        ;AGAIN. IT IS CHECKED IF IT WAS A MISS.
                                         ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                ;;*****************************************************************
044224  000004                  TST44:  SCOPE
044226  005005                  VSG0:   CLR     R5
044230  010537  177746          VSG0A:  MOV     R5, @#CONTRL
044234  032737  010000  177746          BIT     #VCIP, @#CONTRL
044242  001374                          BNE     .-6
044244  012702  000034                  MOV     #S0M0M1,R2
044250  012703  000054                  MOV     #S1M0M1,R3
044254  050502                          BIS     R5,R2
044256  050503                          BIS     R5,R3
044260  012700  044226                  MOV     #VSG0, R0       ;MAKE TEST-CODE HIT IN
044264  012701  001000                  MOV     #1000, R1       ;GROUP 1
044270  010237  177746          1$:    MOV     R2, @#CONTRL;FORCE REPLACE GROUP 0
044274  005762  002000                  TST     2000(R2)
044300  010337  177746                  MOV     R3, @#CONTRL    ;FORCE REPLACE GROUP 1
044304  005720                          TST     (R0)+
044306  077110                          SOB     R1, 1$
044310  012700  114704                  MOV     #TSTDAT, R0     ;MAKE TEST-DATA HIT IN
044314  012701  001000                  MOV     #1000, R1       ;GROUP 0
044320  010337  177746                  MOV     R3, @#CONTRL
044324  042737  000014  177746          BIC     #M0M1, @#CONTRL   ;FORCE REPLACE GROUP 0
044332  005720                  2$:     TST     (R0)+
044334  077102                          SOB     R1, 2$
044336  042737  000020  177746          BIC     #S0, @#CONTRL
044344  052737  000040  177746          BIS     #S1, @#CONTRL   ;FORCE REPLACE GROUP 1
044352  052737  000400  177746          BIS     #FCAC,@#CONTRL  ;FLUSH CACHE
044360  013704  177746                  MOV     @#CONTRL, R4
044364  074504                          XOR     R5, R4          ;CHECK IF VSIU COMPLEMENTED
044366  032704  020000                  BIT     #VSIU, R4
044372  001004                          BNE     3$
```

```
044374  013737  177746  001556          MOV     @#CONTRL,$REG0
044402  104064                          ERROR   64                      ;VSIU DID NOT SWITCH WHEN
                                                                        ;CACHE-FLUSH WAS DONE
044404  052737  000014  177746  3$:     BIS     #MOM1, @#CONTRL ;MAKE TEST-CODE HIT IN
044412  012700  044226                  MOV     #VSG0, R0        ;GROUP 1
044416  012701  001000                  MOV     #1000,R1
044422  005720                  4$:     TST     (R0)+
044424  077102                          SOB     R1, 4$
044426  042737  000014  177746          BIC     #MOM1, @#CONTRL
044434  012700  114704                  MOV     #TSTDAT, R0      ;REFERENCE TEST-DATA AND CHECK
044440  012701  000400                  MOV     #400, R1        ;THAT IT IS A MISS. NOTE
044444  005710                  5$:     TST     (R0)            ;SETTING CACHE-FLUSH SHOULD
044446  032737  000010  177752          BIT     #10, @#HITMIS    ;HAVE INVALIDATED GROUP 0
044454  001410                          BEQ     6$
044456  013737  177746  001556          MOV     @#CONTRL,$REG0
044464  005037  001560                  CLR     $REG1            ;GROUP NO.
044470  010037  001562                  MOV     R0,$REG2         ;TEST DATA ADDRESS
044474  104066                          ERROR   66              ;TEST-DATA WAS NOT A MISS.
                                                                ;TEST DATA WAS MADE A HIT
                                                                ;IN GROUP 0 AND THEN CACHE-
                                                                ;FLUSH WAS DONE. CACHE-FLUSH
                                                                ;SHOULD HAVE INVALIDATED GROUP
                                                                ;0'S CACHED DATA, HENEE, THE
                                                                ;TEST DATA REFERENCE SHOULD
                                                                ;HAVE BEEN A MISS.
                                                                ;PROBLE FAULURE
044476  062700  000004          6$:     ADD     #4, R0          ;VALID STORE IS NOT BEING SWITCHED
044502  077120                          SOB     R1, 5$          ;TO THE OTHER WHEN CACHE-FLUSH IS
                                                                ;SET IN THE CCR
044504  032737  010000  177746  7$:     BIT     #VCIP, @#CONTRL
044512  001374                          BNE     7$
044514  012700  020000                  MOV     #VSIU, R0        ;COMPLEMENT VSIU
044520  074037  177746                  XOR     R0, @#CONTRL
044524  032737  010000  177746  8$:     BIT     #VCIP, @#CONTRL
044532  001374                          BNE     8$
044534  052737  000014  177746          BIS     #MOM1, @#CONTRL ;MAKE TEST-CODE HIT IN
044542  012700  044226                  MOV     #VSG0, R0        ;GROUP 1
044546  012701  001000                  MOV     #1000, R1
044552  005720                   9$:    TST     (R0)+
044554  077102                          SOB     R1, 9$
044556  042737  000014  177746          BIC     #MOM1, @#CONTRL
                                                                ;THE ORIGINAL VALID STORE (WHICH
                                                                ;WAS INVALIDATED BY CACHE FLUSH)
                                                                ;IS IN USE AGAIN.
044564  012700  114704                  MOV     #TSTDAT, R0
044570  012701  000400                  MOV     #400, R1        ;REFERENCE THE TEST-DATA AND
                                                                ;CHECK IT IS A MISS
044574  005710                  10$:    TST     (R0)
044576  032737  000010  177752          BIT     #10, @#HITMIS
044604  001410                          BEQ     11$
044606  013737  177746  001556          MOV     @#CONTRL,$REG0
044614  005037  001560                  CLR     $REG1            ;GROUP NO.
044620  010037  001562                  MOV     R0,$REG2         ;TEST DATA ADDRESS
044624  104067                          ERROR   67        ;TEST-DATA REFERENCE WAS NOT A MISS (IN
                                                         ;GROUP 0, ORIGINAL VALID STORE). CACHE-FLUSH
                                                         ;DONE EARLIER ON THE ORIGINAL VALID STORE
```

```
                                                                 ;SHOULD HAVE RESULTED IN INVALIDATING
                                                                 ;THE VALID STORE, THUS RESULTING IN
                                                                 ;CACHE-MISS ON TEST DATA REFERENCE.
                                                                 ;PROBALE FAILURE: VALID STORE IN USE IS NOT
                                                                 ;BEING INVALIDATD WHEN CACHE-FLUSH IS
                                                                 ;SET
044626  062700  000004          11$:     ADD     #4, R0
044632  077120                           SOB     R1, 10$
044634  012701  020000                   MOV     #VSIU,R1
044640  074105                           XOR     R1,R5    ;TESTED VALID STORE B (1)?
044642  001402                           BEQ     TST45             ;;EXIT
044644  000137  044230                   JMP     VSG0A



                                ;;*******************************************************************
                                ;*TEST 45        CHECK VALID STORES (A&B) FOR GROUPES 0 & 1
                                         ;THIS TEST CHECKS THAT HIT CAN BE OBTAINED FROM BOTH GROUPS
                                         ;(0&1) OF THE CACHE, FROM EACH OF THE TWO VALID
                                         ;STORES (A&B) PER GROUP. THUS ALL 4 VALID STORES GET
                                         ;CHECKED.
                                         ;TEST-DATA (UNIQUE) IS MADE A HIT IN GROUP 0 USING
                                         ;THE FIRST VALID STORE A. TEST-CODE IS MADE A HIT IN THE
                                         ;GROUP NOT BEING TESTED. TEST-DATA IS READ BACK AND
                                         ;CHECKED FOR CORRECTNESS. IT IS ALSO CHECKED IF THE
                                         ;TEST-DATA REFERENCE WAS A HIT. THE TESTING !S
                                         ;REPEATED FOR VALID STORE B.
                                         ;THE ENTIRE TEST (ABOVE) IS REPEATED FOR GROUP 1.
                                          ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                ;;*******************************************************************
044650  000004                  TST45:   SCOPE
044652  005002                  G1G0V:   CLR     R2         ;VSIU HIT MASK
044654  005005                           CLR     R5         ;INITIALIZE COUNT DATA PATTERN TO BE USED
044656  012700  000034          G1G0VA:  MOV     #S0M0M1, R0
044662  012701  000054                   MOV     #S1M0M1, R1
044666  010237  177746          G1G0VB:  MOV     R2, @#CONTRL
044672  032737  010000  177746  1$:      BIT     #VCIP,@#CONTRL
044700  001374                           BNE     1$
044702  050200                           BIS     R2,R0
044704  050201                           BIS     R2,R1
044706  012703  044652                   MOV     #G1G0V, R3         ;MAKE TEST-CODE HIT IN THE
044712  012704  001000                   MOV     #1000, R4          ;GROUP NOT BEINNG TESTED
044716  010037  177746          2$:      MOV     R0, @#CONTRL
044722  005763  002000                   TST     2000 (R3)
044726  010137  177746                   MOV     R1, @#CONTRL
044732  005723                           TST     (R3)+
044734  077410                           SOB     R4, 2$
044736  042700  000014                   BIC     #M0M1, R0          ;WRITE COUNT PATTERN AND MAKE
044742  042701  000014                   BIC     #M0M1, R1          ;IT A HIT IN THE GROUP BEING
044746  012703  114704                   MOV     #TSTDAT, R3        ;TESTED.
044752  012704  001000                   MOV     #1000, R4          ;BIT 15 OF THE COUNT PATTERN INDICATES
                                                                    ;WHICH GROUP;BIT15=0, GROUP 0, ELSE 1
044756  010037  177746                   MOV     R0,@#CONTRL        ;BIT 14 OF THE COUNT PATTERN INDICATES
044762  010513                  3$:      MOV     R5, (R3)          ;WHICH VALID STORE, A (0) OR B (1)
044764  005723                           TST     (R3)+              ;MAKE IT A HIT
044766  005205                           INC     R5
044770  077404                           SOB     R4, 3$
```

```
044772  010137  177746                  MOV     R1, @#CONTRL
044776  012703  114704                  MOV     #TSTDAT, R3
045002  012704  001000                  MOV     #1000,  R4
045006  042705  001777                  BIC     #1777, R5          ;INITIALIZE PATTERN TO BE CHECKED
045012  011337  045166          4$:     MOV     (R3), TMP         ;READ THE TEST-DATA AND
045016  032737  000020  177752          BIT     #20, @#HITMIS     ;CHECK IF THE REFERENCE WAS
045024  001016                          BNE     5$                ;A HIT
045026  013737  177746  001556          MOV     @#CONTRL,$REG0
045034  005037  001560                  CLR     $REG1              ;GROUP NO.
045040  032705  100000                  BIT     #BIT15,R5          ;WHICH GROUP?
045044  001403                          BEQ     8$
045046  012737  000001  001560          MOV     #1,$REG1
045054  010337  001562          8$:     MOV     R3,$REG2           ;TEST DATA ADDRESS
045060  104070                          ERROR   70                ;TEST-DATA REFERENCE WAS NOT A
                                                                  ;HIT, FROM THE GROUP AND
                                                                  ;VALID STORE BEING TESTED
045062  023705  045166          5$:     CMP     TMP, R5           ;DATA CORRECT?
045066  001410                          BEQ     6$
045070  010537  001556                  MOV     R5,$REG0           ;EXPCTD DATA
045074  013737  045166  001560          MOV     TMP,$REG1          ;DATA RECVD
045102  010337  001562                  MOV     R3,$REG2
045106  104071                          ERROR   71                 ;READ INCORRECT DATA ON REFEREN
                                                                   ;-CING A CACHED LOCATION.
045110  062703  000002          6$:     ADD     #2, R3
045114  005205                          INC     R5
045116  077443                          SOB     R4, 4$
045120  012704  020000                  MOV     #VSIU,R4
045124  074402                          XOR     R4, R2          ;DONE VALID STORE B (1)?
045126  001405                          BEQ     7$                 ;YES
045130  052705  040000                  BIS     #BIT14, R5        ;INDICATE VS-B IN DATA-PATTERN
045134  042705  001777                  BIC     #1777,R5
045140  000646                          BR      G1G0VA             ;CHECK GROUP, VS-B
045142  032705  100000          7$:     BIT     #BIT15, R5         ;DONE CHECKING GROUP 1?
045146  001010                          BNE     TST46              ;;EXIT
045150  012700  000054                  MOV     #S1M0M1, R0
045154  012701  000034                  MOV     #S0M0M1, R1
045160  012705  100000                  MOV     #BIT15, R5 ;INDICATE GROUP 1
045164  000640                          BR      G1G0VB
045166  000000                  TMP:    .WORD   0



                                ;;*******************************************************************
                                ;*TEST 46         CHECK VALID STORES (A &B ) FOR GROUP 1
                                        ;THIS TEST CHECKS RTHE TWO VALID STORES (A&B) FOR GROUP 1
                                        ;OF THE CACHE. WHEN A CACHE-FLUSH IS ISSUED, THE CACHE
                                        ;SHOULD BE INVALIDATED BY SWITCHING THE VALID STORE
                                        ;IN USE.
                                        ;THE TEST-CODE IS MADE HIT IN GROUP 1 (WHICH IS NOT
                                        ;BEING TESTED). THE TEST DATA IS MADE HIT IN GROUP 0.
                                        ;FLUSH-CACHE HIT IS SET IN THE CCR. IT IS CHECKED THAT
                                        ;THE TEST-DATA WHICH WAS HIT (MADE PRECIOUSLY) IN
                                        ;GROUP 0 IS NO MORE A HIT, EACH LOCATION OF THE
                                        ;TEST-DATA BLOCK IS REFERENCED AND CHECKED IF
                                        ;IT WAS A MISS. OTHERWISE AN ERROR IS REPORTED. AS A
                                        ;RESULT OF THE CACHE FLUSH THE VALID STORE SHOULD
                                        ;HAVE SWITCHED FROM 0 TO 1. THEN THE VALID STORE
                                        ;IS FORCED TO BE 0 AND THE TEST-DATA IS REFERENCED
                                        ;AGAIN. IT IS CHECKED IF IT WAS A MISS.
                                        ;THE WHOLE TEST IS REPEATED USING VALID-STORE
                                        ;B (1).
                                         ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                ;;******************************************************************
045170  000004                  TST46:  SCOPE
045172  005005                  VSG1:   CLR     R5                ;R5, HIT MASK FOR VSIU
045174  010537  177746          VSG1A:  MOV     R5, @#CONTRL
045200  032737  010000  177746          BIT     #VCIP,@#CONTRL
045206  001374                          BNE     .-6
045210  012702  000034                  MOV     #S0M0M1,R2
045214  012703  000054                  MOV     #S1M0M1,R3
045220  050502                          BIS     R5,R2
045222  050503                          BIS     R5,R3
045224  012700  045172                  MOV     #VSG1, R0         ;MAKE TEST-CODE HIT IN
045230  012701  001000                  MOV     #1000, R1          ;GROUP 0
045234  010337  177746          1$:     MOV     R3, @#CONTRL;FORCE REPLACE GROUP 1
045240  005760  002000                  TST     2000(R0)
045244  010237  177746                  MOV     R2, @#CONTRL      ;FORCE REPLACE GROUP 0
045250  005720                          TST     (R0)+
045252  077110                          SOB     R1, 1$
045254  012700  114704                  MOV     #TSTDAT, R0       ;MAKE TEST-DATA HIT IN
045260  012701  001000                  MOV     #1000, R1          ;GROUP 1
045264  010337  177746                  MOV     R3, @#CONTRL      ;FORCE REPLACE GROUP 1
045270  042737  000014  177746          BIC     #M0M1,@#CONTRL
045276  005720                  2$:     TST     (R0)+
045300  077102                          SOB     R1, 2$
045302  042737  000040  177746          BIC     #S1, @#CONTRL
045310  052737  000020  177746          BIS     #S0, @#CONTRL     ;FORCE REPLACE GROUP 0
045316  052737  000400  177746          BIS     #FCAC, @#CONTRL   ;FLUSH CACHE
045324  013704  177746                  MOV     @#CONTRL, R4
045330  074504                          XOR     R5, R4            ;CHECK IF VSIU COMPLEMENTED
045332  032704  020000                  BIT     #VSIU, R4
045336  001004                          BNE     3$
045340  013737  177746  001556          MOV     @#CONTRL,$REG0
045346  104064                          ERROR   64                ;VSIU DID NOT SWITCH WHEN
                                                                  ;CACHE-FLUSH WAS DONE
045350  052737  000014  177746  3$:     BIS     #M0M1,  @#CONTRL;MAKE TEST-CODE HIT IN
045356  012700  045172                  MOV     #VSG1,R0                   ;GROUP0
045362  012701  001000                  MOV     #1000,R1
045366  005720                  4$:     TST     (R0)+
045370  077102                          SOB     R1, 4$
045372  042737  000014  177746          BIC     #M0M1,@#CONTRL
045400  012700  114704                  MOV     #TSTDAT, R0        ;REFERENCE TEST-DATA AND CHECK
045404  012701  000400                  MOV     #400,R1           ;THAT IT IS A MISS. NOTE
045410  005710                  5$:     TST     (R0)              ;SETTING CACHE-FLUSH SHOULD
045412  032737  000010  177752          BIT     #10, @#HITMIS      ;HAVE INVALIDATED GROUP
045420  001411                          BEQ     6$
045422  013737  177746  001556          MOV     @#CONTRL,$REG0
045430  012737  000001  001560          MOV     #1,$REG1           ;GROUP NO.
045436  010037  001562                  MOV     R0,$REG2           ;TEST DATA ADDRESS
045442  104066                          ERROR   66                ;TEST-DATA WAS NOT A MISS.
                                                                  ;TEST DATA WAS MADE A HIT
                                                                  ;IN GROUP 1 AND THEN CACHE-
                                                                  ;FLUSH WAS DONE. CACHE-FLUSH
```

```
                                                                         ;SHOULD HAVE INVALIDATED GROUP
                                                                         ;1'S CACHED DATA. HENEE, THE
                                                                         ;TEST DATA REFERENCE SHOULD
                                                                         ;HAVE BEEN A MISS.
                                                                         ;PROBABLE FAILURE:
045444  062700  000004          6$:     ADD     #4, R0                   ;VALID STORE IS NOT BEING SWITCHED
045450  077121                          SOB     R1, 5$                   ;TO THE OTHER WHEN CACHE-FLUSH IS
                                                                         ;SET IN THE CCR
045452  032737  010000  177746  7$:     BIT     #VCIP, @#CONTRL
045460  001374                          BNE     7$
045462  012700  020000                  MOV     #VSIU, R0                ;COMPLEMENT VSIU
045466  074037  177746                  XOR     R0, @#CONTRL
045472  032737  010000  177746  8$:     BIT     #VCIP, @#CONTRL
045500  001374                          BNE     8$
045502  052737  000014  177746          BIS     #MOM1, @#CONTRL ;MAKE TEST-CODE HIT IN
045510  012700  045172                  MOV     #VSG1, R0                ;GROUP 0
045514  012701  001000                  MOV     #1000, R1
045520  005720                  9$:     TST     (R0)+
045522  077102                          SOB     R1, 9$
045524  042737  000014  177746          BIC     #MOM1, @#CONTRL
                                                                         ;THE ORIGINAL VALID STORE (WHICH
                                                                         ;WAS INVALIDATED BY CACHE FLUSH)
                                                                         ;IS IN USE AGAIN.
045532  012700  114704                  MOV     #TSTDAT, R0
045536  012701  000400                  MOV     #400, R1                 ;REFERENCE THE TEST-DATA AND
                                                                         ;CHECK IT IS A MISS
045542  005710                  10$:    TST     (R0)
045544  032737  000010  177752          BIT     #10, @#HITMIS
045552  001411                          BEQ     11$
045554  013737  177746  001556          MOV     @#CONTRL,$REG0
045562  012737  000001  001560          MOV     #1,$REG1                 ;GROUP NO.
045570  010037  001562                  MOV     R0,$REG2                 ;TEST DATA ADDRESS
045574  104067                          ERROR   67              ;TEST-DATA REFERENCE WAS NOT A MISS (IN
                                                               ;GROUP 1, ORIGINAL VALID STORE). CACHE-FLUSH
                                                               ;DONE EARLIER ON THE ORIGINAL VALID STORE
                                                               ;SHOULD HAVE RESULTED IN INVALIDATING
                                                               ;THE VALID STORE, THUS RESULTING IN
                                                               ;CACHE-MISS ON TEST DATA REFERENCE.
                                                               ;PROBABLE FAILURE: VALID STORE IN USE IS NOT
                                                               ;BEING INVALIDATED WHEN CACHE-FLUSH IS
                                                               ;SET
045576  062700  000004          11$:    ADD     #4, R0
045602  077121                          SOB     R1, 10$
045604  012701  020000                  MOV     #VSIU,R1
045610  074105                          XOR     R1, R5 ;TESTED VALID STORE B (1)?
045612  001402                          BEQ     TST47           ;;EXIT
045614  000137  045174                  JMP     VSG1A


                                ;;*****************************************************************
                                ;*TEST 47        CHECK CACHE TURNS OFF WHEN FLUSH IS DONE WITH IVSS SET
                                        ;THIS TEST CHECKS THAT IF CACHE-FLUSH IS DONE (SETTING CF), WHEN IVSS
                                        ;IS SET, THE VALID STORES ARE NOT SWITCHED AND THE CACHE IS TURNED
                                        ;OFF (AND A SLOW FLUSH IS PERFORMED). THUS, ANY REFERENCE TO
                                        ;A PREVIOUSLY CACHED DATA SHOULD RESULT IN CACHE MISS.
                                        ;TEST-DATA IS MADE HIT IN GROUP 0 (BEING TESTED). TEST CODE IS
```

```
                                            ;MADE HIT IN GROUP 1.IVSS IS SET AND A FLUSH IS DONE. PREVIOUSLY
                                            ;CACHED TEST-DATA IS REFERENCED TO CHECK IT IS A MISS.
                                            ;THE TEST IS REPEATED FOR BOTH GROUPS AND VALID STORES.
                                            ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                    ;;*******************************************************************
045620  000004                      TST47:  SCOPE

045622  005002                      IVFC:   CLR     R2                      ;BIT MASK FOR VSIU
045624  012700  000034                      MOV     #S0M0M1,R0
045630  012701  000054                      MOV     #S1M0M1,R1
045634  050200                      IVFCA:  BIS     R2,R0
045636  050201                              BIS     R2,R1
045640  010237  177746                      MOV     R2,@#CONTRL
045644  032737  010000  177746      1$:     BIT     #VCIP,@#CONTRL
045652  001374                              BNE     1$

045654  012703  045622                      MOV     #IVFC,R3                        ;MAKE TEST CODE BIT IN GROUP
045660  012704  001000                      MOV     #1000,R4                ;NOT BEING TESTED
045664  010037  177746              2$:     MOV     R0,@#CONTRL
045670  005763  002000                      TST     2000(R3)
045674  010137  177746                      MOV     R1,@#CONTRL
045700  005723                              TST     (R3)+
045702  077410                              SOB     R4,2$

045704  042700  000014                      BIC     #M0M1,R0
045710  042701  000014                      BIC     #M0M1,R1

045714  012703  114704                      MOV     #TSTDAT,R3              ;MAKE TEST-DATA HIT IN GROUP
045720  012704  001000                      MOV     #1000,R4                ;BEING TESTED

045724  010037  177746                      MOV     R0,@#CONTRL
045730  005723                      3$:     TST     (R3)+
045732  077402                              SOB     R4, 3$

045734  010137  177746                      MOV     R1,@#CONTRL              ;FORCE REPLACE GROUP (NOT BEING TESTED)
045740  052737  040000  177746              BIS     #IVSS,@#CONTRL          ;SET IVSS

045746  012705  000004                      MOV     #4,R5                   ;BIT MASK FOR HIT/MISS REGISTER
045752  012704  000400                      MOV     #400,R4
045756  012703  116704                      MOV     #TSTDAT+2000,R3
045762  052737  000400  177746              BIS     #FCAC,@#CONTRL          ;FLUSH CACHE
045770  005743                      4$:     TST     -(R3)                   ;REFERENCE PREVIOUSLY CACHED
045772  030537  177752                      BIT     R5,@#HITMIS             ;TEST DATA& CHECK IT IS A MISS
045776  001004                              BNE     6$
046000  162703  000002              5$:     SUB     #2,R3
046004  077407                              SOB     R4,4$
046006  000417                              BR      7$                      ;DONE

046010  013737  177746  001556      6$:     MOV     @#CONTRL,$REG0
046016  005037  001560                      CLR     $REG1                   ;GROUP NO.
046022  032700  000040                      BIT     #S1,R0                  ;WHICH GROUP?
046026  001403                              BEQ     12$
046030  012737  000001  001560              MOV     #1,$REG1                ;GROUP NO
046036  010337  001562              12$:    MOV     R3,$REG2                ;TEST DATA ADDRESS
046042  104072                              ERROR   72                      ;TEST DATA REFERENCE DID NOT
                                                                            ;REGISTER A MISS. TEST-DATA WAS
```

```
046044  000755                          BR      5$              ;MADE BIT IN A GROUP. CACHE-FLUSH
                                                                ;WAS DONE, WITH IVSS SET. REFERENCE
                                                                ;TO THE PREVIOUSLY CACHED DATA
                                                                ;SHOULD HAVE BEEN A MISS.
                                                                ;PROBABLE FAILURE:  CACHE DOES NOT
                                                                ;TURN OFF WHEN IVSS IS SET AND
                                                                ;FLUSH IS PERFORMED
                                        ;CHECK THAT THE CACHE HAS TURNED ON AGAIN,CHECK
                                        ;THAT HITS CAN BE OBTAINED

046046  012703  045622          7$:     MOV     #IVFC,R3        ;MAKE THE TEST-CODE HIT IN GROUP NOT
046052  012704  001000                  MOV     #1000,R4        ;BEING TESTED
046056  052700  000014                  BIS     #MOM1,R0
046062  052701  000014                  BIS     #MOM1,R1
046066  010037  177746          8$:     MOV     R0,@#CONTRL
046072  005763  002000                  TST     2000(R3)

046076  010137  177746                  MOV     R1,@#CONTRL
046102  005723                          TST     (R3)+
046104  077410                          SOB     R4,8$

046106  042700  000014                  BIC     #MOM1,R0
046112  042701  000014                  BIC     #MOM1,R1

046116  012703  114704                  MOV     #TSTDAT,R3      ;MAKE TEST-DATA HIT IN GROUP
046122  012704  001000                  MOV     #1000,R4        ;BEING TESTED

046126  010037  177746                  MOV     R0,@#CONTRL
046132  005723                  9$:     TST     (R3)+
046134  077402                          SOB     R4,9$

                                                                ;FORCE REPLACE GROUP NOT BEING
046136  010137  177746                  MOV     R1,@#CONTRL     ;TESTED
046142  012703  114704                  MOV     #TSTDAT,R3      ;REFERENCE TEST-DATA (IN THE
046146  012704  001000                  MOV     #1000,R4        ;GROUP BEING CHECKED) AND
046152  005713                  10$:    TST     (R3)            ;MAKE SURE IT IS A HIT
046154  032737  000010  177752          BIT     #10,@#HITMIS    ;HIT?
046162  001016                          BNE     11$             ;YES
046164  013737  177746  001556          MOV     @#CONTRL,$REG0
046172  005037  001560                  CLR     $REG1           ;GROUP NO.
046176  032700  000040                  BIT     #S1,R0          ;WHICH GROUP?
046202  001403                          BEQ     13$
046204  012737  000001  001560          MOV     #1,$REG1        ;GROUP NO
046212  010337  001562          13$:    MOV     R3,$REG2        ;TEST DATA ADDRESS
046216  104073                          ERROR   73              ;PREVIOUSLY CACHED TEST-DATA
                                                                ;WAS REFERENCED BUT IT
                                                                ;WAS NOT A HIT.
                                                                ;POSSIBLE FAULT: CACHE DID NOT
                                                                ;TURN ON AFTER HAVING TURNED
                                                                ;OFF (WHEN A CACHE FLUSH
                                                                ;WAS DONE WITH IVSS SET).

046220  062703  000002          11$:    ADD     #2,R3
046224  077426                          SOB     R4,10$          ;DONE?
046226  052700  000014                  BIS     #MOM1,R0
046232  052701  000014                  BIS     #MOM1,R1
```

```
046236  012704  020000              MOV     #VSIU,R4
046242  074402                      XOR     R4,R2           ;DONE VALID STORE B?
046244  001402                      BEQ     14$
046246  000137  045634              JMP     IVFCA
046252  032700  000040      14$:    BIT     #S1,R0          ;YES, DONE GROUP 1?
046256  001007                      BNE     TST50           ;;YES,EXIT
```

```
046260  012700  000054                   MOV     #S1M0M1,R0         ;CCR MASKS FOR GROUP 1 TESTING
046264  012701  000034                   MOV     #S0M0M1,R1
046270  005002                           CLR     R2                 ;BIT MASK FOR VALID STORE
046272  000137  045634                   JMP     IVFCA              ;GO TEST GROUP 1



                                 ;;*****************************************************************
                                 ;*TEST 50        CHECK CACHE TURNS OFF ON A BACK-TO-BACK FLUSH
                                         ;THIS TEST CHECKS THAT THE CACHE TURNS OFF AND FORCES
                                         ;ALL REFERENCES TO THE MAIN MEMORY WHEN BACK-TO-BACK
                                         ;CACHE FLUSHES ARE DONE. WHEN A CACHE FLUSH IS INITIATED
                                         ;WHILE THE PREVIOUS ONE IS IN PROGRESS, IT IS KNOWN
                                         ;AS BACK-TO-BACK FLUSH.
                                         ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                 ;;*****************************************************************
046276  000004                   TST50:  SCOPE
046300  005037  177746                   CLR     @#CONTRL
046304  032737  010000  177746   1$:     BIT     #VCIP,@#CONTRL
046312  001374                           BNE     1$
046314  012701  177774                   MOV     #-4,R1
046320  012700  000007                   MOV     #7,R0
046324  012737  000400  177746           MOV     #FCAC,@#CONTRL   ;FLUSH CACHE
046332  012737  000400  177746           MOV     #FCAC,@#CONTRL   ;AGAIN FLUSH THE CACHE. SINCE
                                                                  ;PREVIOUS FLUSH IS STILL IN
046340  005201                   2$:     INC     R1                 ;PROGRESS CACHE SHOULD BE
046342  001410                           BEQ     3$                 ;TURNED OFF
046344  030037  177752                   BIT     R0,@#HITMIS        ;CHECK THAT THE LAST THREE REFERENCES
046350  001773                           BEQ     2$                 ;WERE MISSES
046352  013702  177752                   MOV     @#HITMIS,R2
046356  010237  001556                   MOV     R2,$REG0
046362  104115                           ERROR   115                ;CACHE DID NOT TURN OFF ON
                                                                    ;PERFORMING A BACK-TO-BACK
                                                                    ;FLUSH. FOR THE PERIOD OF TIME
                                                                    ;THAT THE FLUSH IS BEING DONE
                                                                    ;AND THE CACHE IS OFF, ALL
                                                                    ;REFERENCES SHOULD BE FORCED
                                                                    ;TO MAIN MEMORY (MISSES).
046364                           3$:                                ;EXIT



                                 ;;*****************************************************************
                                 ;*TEST 51        CHECK CACHE-BYPASS
                                         ;THIS TEST CHECKS THE CACHE BYPASS FUNCTION.  WHEN THE
                                         ;'BYPASS CACHE'' IS SET IN THE CACHE CONTROL REGISTER
                                         ;ALL REFERENCES ARE FORCED TO MAIN MEMORY. IF A
                                         ;READ OR WRITE HIT OCCURS THAT LOCATION IS INVAL-
                                         ;-IDATED IN THE TAG STORE.
                                         ;FIRST, THE TEST CODE IS MADE HIT IN GROUP 1 BY
                                         ;FORCE-REPLACING GROUP 1. THEN THE TEST-DATA IS MADE
                                         ;HIT IN GROUP 0. CACHE-BYPASS IS SET AND THE TEST
                                         ;DATA (WHICH HAS BEEN CACHED IN GROUP 0) IS
                                         ;REFERENCED.  THE REFERENCES ARE CHECKED FOR MISSES
                                         ;(THE TEST-DATA INSIDE THE CACHE GROUP-0 SHOULD
                                         ;HAVE BEEN INVALIDATED WHEN REFERENCES WERE
                                         ;MADE WITH CACHE-BYPASS SET.)
                                         ;THE ENTIRE TEST IS REPEATED, SELECTING THE
```

```
                                           ;OTHER VALID STORE AND THEN WITH TEST-DATA IN
                                           ;GROUP 1.
                                           ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                   ;;*******************************************************************
046364   000004                    TST51:  SCOPE
046366   005002                    CBP:    CLR     R2                 ;BIT MASK FOR VSIU
046370   012700   000034                   MOV     #S0M0M1,R0
046374   012701   000054                   MOV     #S1M0M1,R1
046400   050200                    CBPA:   BIS     R2,R0
046402   050201                            BIS     R2,R1
046404   010237   177746                   MOV     R2,@#CONTRL
046410   032737   010000   177746  1$:     BIT     #VCIP,@#CONTRL     ;SELECT VSIU
046416   001374                            BNE     1$

046420   012703   046366                   MOV     #CBP,R3            ;MAKE TEST-CODE HIT IN THE
046424   012704   001000                   MOV     #1000,R4           ;GROUP NOT BEING TESTED
046430   010037   177746           2$:     MOV     R0,@#CONTRL
046434   005763   002000                   TST     2000(R3)
046440   010137   177746                   MOV     R1,@#CONTRL
046444   005723                            TST     (R3)+
046446   077410                            SOB     R4,2$

046450   042700   000014                   BIC     #M0M1,R0
046454   042701   000014                   BIC     #M0M1,R1

046460   012703   114704                   MOV     #TSTDAT,R3         ;MAKE TEST-DATA HIT IN THE
046464   012704   001000                   MOV     #1000,R4           ;GROUP BEING TESTED
046470   010037   177746                   MOV     R0,@#CONTRL
046474   005723                    3$:     TST     (R3)+
046476   077402                            SOB     R4,3$
046500   010137   177746                   MOV     R1,@#CONTRL        ;FORCE REPLACE IN THE GROUP NOT
                                                                      ;BEING TESTED
046504   052737   001000   177746          BIS     #UCB,@#CONTRL      ;UNCONDITIONED CACHE BY-PASS

046512   012703   114704                   MOV     #TSTDAT,R3
046516   012704   001000                   MOV     #1000,R4           ;REFERENCE THE CACHED-TEST-DATA
046522   005713                    4$:     TST     (R3)               ;THE GROUP BEING TESTED
046524   032737   000010   177752          BIT     #10,@#HITMIS       ;MISS?
046532   001416                            BEQ     5$                 ;YES

046534   013737   177746   001556          MOV     @#CONTRL,$REG0
046542   005037   001560                   CLR     $REG1              ;GROUP NO.
046546   032700   000040                   BIT     #S1,R0             ;WHICH GROUP?
046552   001403                            BEQ     8$
046554   012737   000001   001560          MOV     #1,$REG1           ;GROUP NO
046562   010337   001562           8$:     MOV     R3,$REG2           ;TEST DATA ADDRESS
046566   104074                            ERROR   74                 ;TEST-DATA-REFERENCE WAS NOT
                                                                      ;A MISS.  TEST-DATA WAS PREVIOUSLY
                                                                      ;CACHED IN THE GROUP BEING
                                                                      ;TESTED.  THEN IT WAS REFERENCED
                                                                      ;WITH CACHE BY-PASS SET.  IT
                                                                      ;SHOULD HAVE BEEN A MISS.
                                                                      ;PROBABLE FAILURE : A MISS IS
                                                                      ;NOT BEING FORCED WHEN CACHE
                                                                      ;BYPASS IS SET
```

```
046570  062703  000002          5$:     ADD     #2,R3
046574  077426                          SOB     R4,4$                   ;DONE?

046576  042737  001000  177746          BIC     #UCB,@#CONTRL           ;CLEAR CACHE BYPASS
046604  012703  114704                  MOV     #TSTDAT,R3              ;REFERENCE THE TEST-DATA AGAIN
046610  012704  000400                  MOV     #400,R4 ;IT SHOULD BE A MISS
046614  005713                  6$:     TST     (R3)
046616  032737  000010  177752          BIT     #10,@#HITMIS            ;MISS?
046624  001416                          BEQ     7$                      ;YES

046626  013737  177746  001556          MOV     @#CONTRL,$REG0
046634  005037  001560                  CLR     $REG1                   ;GROUP NO.
046640  032700  000040                  BIT     #S1,R0                  ;WHICH GROUP?
046644  001403                          BEQ     9$
046646  012737  000001  001560          MOV     #1,$REG1                ;GROUP NO
046654  010337  001562          9$:     MOV     R3,$REG2                ;TEST DATA ADDRESS
046660  104075                          ERROR   75                      ;TEST-DATA-REFERENCE WAS NOT
                                                                        ;A MISS.  TEST-DATA WAS PREVIOUSLY
                                                                        ;CACHED IN THE GROUP BEING
                                                                        ;TESTED.  THEN IT WAS INVALIDATED
                                                                        ;BY REFERENCING IT WHILE
                                                                        ;CACHE-BYPASS WAS SET.  THEN
                                                                        ;CACHE-BYPASS WAS CLEARED AND
                                                                        ;THE TEST DATA WAS REFERENCED
                                                                        ;AGAIN TO MAKE SURE IT WAS
                                                                        ;INVALIDATED.
046662  062703  000004          7$:     ADD     #4,R3                   ;PROBABLE FAILURE - CACHE-BYPASSS
046666  077426                          SOB     R4,6$                   ;DOES NOT INVALIDATE DATE
                                                                        ;THAT IS A HIT INSIDE THE
046670  052700  000014                  BIS     #MOM1,R0                ;CACHE
046674  052701  000014                  BIS     #MOM1,R1
046700  012704  020000                  MOV     #VSIU,R4
046704  074402                          XOR     R4,R2   ;DONE BOTH VALID STORES?
046706  001234                          BNE     CBPA                    ;NO
046710  032700  000040                  BIT     #S1,R0                  ;TESTED GROUP 1
046714  001005                          BNE     TST52                   ;;EXIT

046716  012700  000054                  MOV     #S1MOM1,R0              ;SET UP FOR TESTING GROUP 1
046722  012701  000034                  MOV     #S0MOM1,R1
046726  000624                          BR      CBPA




                                ;;*******************************************************************
                                ;*TEST 52       CHECK CACHE IS BYPASSED ON ASRB OPERAND
                                        ;THIS TEST CHECKS THAT THE CACHE IS BYPASSED ON THE
                                        ;OPERAND OF THE ASRB INSTRUCTION AND ALSO THE OPERAND
                                        ;IS INVALIDATED.  TEST-CODE (INCLDING THE OPERAND
                                        ;OF THE ASRB) IS MADE HIT IN GROUP 1.  THEN
                                        ;ASRB INSTRUCTION IS EXECUTED ON THE CACHED
                                        ;OPERAND.  IT IS CHECKED IF THE REFERENCE TO THE
                                        ;BYTE-OPERAND WAS A MISS.  THEN THE SAME OPERAND
                                        ;REFERENCED USING AN ORDINARY (NON-BYPASSING)
                                        ;INSTRUCTED.  AGAIN, THE REFERENCE IS CHECKED FOR
                                        ;A MISS.
                                        ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                        ;;*****************************************************************
 046730  000004                         TST52:  SCOPE

 046732  012703  046732                 ASRBCB: MOV     #ASRBCB,R3
 046736  012704  001000                         MOV     #1000,R4         ;MAKE TEST-CODE HIT IN GROUP
 046742  012737  000034  177746         1$:     MOV     #S0M0M1,@#CONTRL;1
 046750  005763  002000                         TST     2000(R3)
 046754  012737  000054  177746                 MOV     #S1M0M1,@#CONTRL
 046762  005723                                 TST     (R3)+
 046764  077412                                 SOB     R4,1$
 046766  042737  000014  177746                 BIC     #M0M1,@#CONTRL
                                                                         ;EXECUTE AN ASRB AND REFERENCE
 046774  106237  047076                         ASRB    @#ASLOC          ;THE TEST LOCATION
 047000  032737  000010  177752                 BIT     #10,@#HITMIS
 047006  001412                                 BEQ     2$

 047010  013737  177746  001556                 MOV     @#CONTRL,$REG0
 047016  012737  000001  001560                 MOV     #1,$REG1                 ;GROUP NO.
 047024  012737  047076  001562                 MOV     #ASLOC,$REG2     ;TEST DATA ADDRESS
 047032  104076                                 ERROR   76               ;PREVIOUSLY CACHED TEST-LOCATION
                                                                         ;WHEN REFERENCED USING AN
                                                                         ;ASRB INSTRUCTION WAS NOT
                                                                         ;A MISS.
                                                                         ;PROBABLE FAILURE = ASRB DOES
                                                                         ;NOT FORCE OPERAND-REFERENCE
                                                                         ;TO THE MAIN MEMORY

 047034  005737  047076                 2$:     TST     @#ASLOC          ;REFERENCE THE TEST-LOCATION
 047040  032737  000010  177752                 BIT     #10,@#HITMIS     ;MISS?
 047046  001414                                 BEQ     TST53            ;;EXIT

 047050  013737  177746  001556                 MOV     @#CONTRL,$REG0
 047056  012737  000001  001560                 MOV     #1,$REG1         ;GROUP NO.
 047064  012737  047076  001562                 MOV     #ASLOC,$REG2     ;TEST DATA ADDRESS
 047072  104077                                 ERROR   77               ;BYTE-OPERAND (OF ASRB) WAS
                                                                         ;NOT INVALIDATED WHEN ASRB
                                                                         ;WAS EXECUTED ON A CACHED
                                                                         ;LOCATION
 047074  000401                                 BR      TST53            ;;EXIT

 047076  000000                         ASLOC:  .WORD   0




                                        ;;*****************************************************************
                                        ;*TEST 53        CHECK CACHE VALID STORE PARITY CHECKER
                                                ;THIS TEST FORCES VALID STORE PARITY ERROR IN THE FOUR
                                                ;VALID STORES AND CHECKS THE PARITY CHECKERS.
                                                ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                        ;;*****************************************************************
 047100  000004                         TST53:  SCOPE
 047102  013700  177744                         MOV     @#MSER,R0
 047106  010037  177744                         MOV     R0,@#MSER
 047112  005002                         CVSPE:  CLR     R2               ;BIT MASK FOR VSIU
 047114  012704  000034                         MOV     #S0M0M1,R4       ;SET UP BIT MASKS TO CHECK
 047120  012705  000054                         MOV     #S1M0M1,R5       ;GROUP 1 FIRST
```

```
047124  050204                          CVSPEA: BIS     R2,R4
047126  050205                                  BIS     R2,R5
047130  010237  177746                          MOV     R2,@#CONTRL
047134  032737  010000  177746              1$: BIT     #VCIP,@#CONTRL
047142  001374                                  BNE     1$

047144  010437  177746                          MOV     R4,@#CONTRL
047150  005737  051224                          TST     @#2$+2000
047154  010537  177746                          MOV     R5,@#CONTRL       ;MAKE ''NOP'' HIT IN GROUP BEING
047160  005737  047224                          TST     @#2$              ;TESTED

047164  032704  000020                          BIT     #S0,R4            ;TESTING GROUP  1?
047170  001004                                  BNE     13$               ;YES
047172  042737  000004  177746                  BIC     #M0,@#CONTRL      ;TESTING GROUP 0,FORCE MISS GROUP 1
047200  000403                                  BR      14$
047202  042737  000010  177746  13$:            BIC     #M1,@#CONTRL      ;TESTING GROUP 1,FORCE MISS GROUP 0

047210  012737  047246  000114  14$:            MOV     #3$,@#114         ;SETUP PARITY ERROR TRAP VECTOR

047216  052737  002000  177746                  BIS     #FVPE,@#CONTRL    ;FORCE VALID STORE PARITY ERROR

047224  000240                              2$: NOP                       ;REFERENCE OF THIS INSTRUCTION
                                                                          ;WILL FORCE A VALID STORE
                                                                          ;PARITY ERROR. TRAP TO 114
                                                                          ;SHOULD OCCUR.
047226  042737  002000  177746                  BIC     #FVPE,@#CONTRL    ;USER FVPE IF STILL SET
047234  013737  177746  001556                  MOV     @#CONTRL,$REG0
047242  104103                                  ERROR   103               ;VALID STORE PARITY ERROR WAS
                                                                          ;FORCED BY SETTINGS FVPE IN
                                                                          ;CCR AND MAKING A REFERENCE
                                                                          ;TO A  LOC THAT WAS MADE
                                                                          ;A HIT.  PARITY ERROR TRAP
                                                                          ;DID NOT OCCUR ON DETECTING
                                                                          ;THAT PARITY ERROR IN VALID
047244  000411                                  BR      6$                ;STORE

047246                                      3$:                           ;ENTER HERE IF A PARITY ERROR
                                                                          ;TRAP OCCURS AS EXPECTED
047246  022626                                  CMP     (SP)+, (SP)+      ;POP THE STACK
047250  032737  002000  177746              5$: BIT     #FVPE,@#CONTRL    ;FVPE CLEARED AFTER PARITY ERROR?
047256  001404                                  BEQ     6$
047260  013737  177746  001556                  MOV     @#CONTRL,$REG0
047266  104105                                  ERROR   105               ;FVPE DID NOT GET CLEARED
                                                                          ;AFTER VALID SOTRE PARITY ERROR
                                                                          ;OCCURED

047270  032737  100000  177746              6$: BIT     #VSPE,@#CONTRL    ;DID-VALID-STORE-PARITY-ERROR SET?
047276  001004                                  BNE     7$
047300  013737  177746  001556                  MOV     @#CONTRL,$REG0
047306  104106                                  ERROR   106               ;VALID-STORE-PARITY-ERROR BIT
                                                                          ;DID NOT SET IN CCR WHEN
                                                                          ;PARITY ERROR FROM V-STORE)
047310  005003                              7$: CLR     R3
                                                                          ;WAS FORCED
047312  032705  000040                          BIT     #S1,R5            ;TESTING GROUP 1?
047316  001003                                  BNE     8$                ;YES
```

```
047320  012700  000020                  MOV     #BIT4,R0                ;SET BIT MASK FOR GROUP 0
047324  000402                          BR      9$
047326  012700  000040             8$:  MOV     #BIT5,R0                ;SET BIT MASK FOR GROUP 1
047332  013701  177744             9$:  MOV     @#MSER,R1               ;GROUP IN WHICH THE PARITY ERROR
047336  042701  177716                  BIC     #^CBIT4+^CBIT5,R1;OCCURED) WAS SET IN MEMORY
047342  020100                          CMP     R1,R0                   ;SYSTEM ERROR REGISTER
047344  001413                          BEQ     10$
047346  013737  177746  001556          MOV     @#CONTRL,$REG0
047354  013737  177744  001560          MOV     @#MSER,$REG1
047362  010337  001562                  MOV     R3,$REG2                ;GROUP NO. BEING TESTED
047366  010037  001564                  MOV     R0,$REG3                ;EXPECTED BITS (4,5)IN MEM SY
                                                                        ;ERROR REGISTER
047372  104107                          ERROR   107                     ;PROPER BITS (ADDRESS MEMORY)
                                                                        ;PARITY ERROR.4,5) WERE NOT
                                                                        ;SET IN MEM-SYS.ERROR
                                                                        ;REGISTER WHEN VALID STORE
                                                                        ;PARITY ERROR WAS FORCED.

047374  013703  177746            10$:  MOV     @#CONTRL,R3             ;DID VSIU BIT SWITCH AFTER
047400  074203                          XOR     R2,R3                   ;PARITY ERROR TRAP?
047402  032703  020000                  BIT     #VSIU,R3
047406  001404                          BEQ     11$                     ;NO, OK
047410  013737  177746  001556          MOV     @#CONTRL,$REG0
047416  104110                          ERROR   110                     ;VSIU  SWITCHED WHEN A VALID
                                                                        ;STORE PARITY ERROR WAS
                                                                        ;FORCED.  IT SHOULD NOT.
047420  013701  177744            11$:  MOV     @#MSER,R1               ;CLEAR MEMORY SYSTEM ERROR
047424  010137  177744                  MOV     R1,@#MSER               ;REGISTER
047430  005737  177744                  TST     @#MSER                  ;CLEARED?
047434  001404                          BEQ     12$
047436  013737  177744  001556          MOV     @#MSER,$REG0
047444  104111                          ERROR   111                     ;MEMORY SYSTEM ERROR REGISTER
                                                                        ;WOULD NOT BE CLEARED BY
                                                                        ;WRITING ITS CONTENTS BACK
                                                                        ;INTO ITSELF.  NOTE PREVIOUSLY
                                                                        ;A VALID STORE PARITY ERROR
                                                                        ;OCCURED THIS SETTINGS ?
                                                                        ;BIT 4 OR 5 IN IT.
047446  012700  020000            12$:  MOV     #VSIU,R0
047452  074002                          XOR     R0,R2                   ;DONE VALID STORE B?
047454  001223                          BNE     CVSPEA                  ;GO CHECK VALID STORE B

047456  032705  000020                  BIT     #S0,R5                  ;CHECKED GROUP 0?
047462  001007                          BNE     TST54                   ;;EXIT

047464  012704  000054                  MOV     #S1M0M1,R4              ;SET UP BIT MASKS TO CHECK
047470  012705  000034                  MOV     #S0M0M1,R5              ;GROUP 0
047474  005002                          CLR     R2                      ;BUT MASK FOR VALID STORE
047476  000137  047124                  JMP     CVSPEA
                                  ;;*************************************************************
                                  ;;*TEST 54              CHECK THAT CACHE-MISS OCCURS ON A VALID STORE PARITY ERROR
                                        ;THIS TEST FORCES A VALID STORE PARITY ERROR AND CHECKS
                                        ;THAT A MISS OCCURS ON THE REFERENCE THAT CAUSED
                                        ;THE PARITY ERROR.  THE CACHE LOCATION THAT GAVE THE
                                        ;PARITY ERROR IS INVALIDATED AND A SLOW CYCLE IS
                                        ;PERFORMED TO THE MAIN MEMORY.  THIS TEST IS
```

```
                                                   ;PERFORMED WITH THE 'DISABLE TRAPS'' BIT OF THE
                                                   ;CACHE CONTROL REGISTER SET, THUS A PARITY ERROR
                                                   ;TRAP WILL NOT OCCUR.  THIS IS DONE SO THAT THE
                                                   ;HIT-MISS REGISTER CAN BE READ WITHOUT LOSING
                                                   ;THE INFORMATION CONTAINED IN IT.
                                                   ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                           ;;*******************************************************************
047502  000004                             TST54:  SCOPE
047504  005002                             VSCM:   CLR     R2                      ;BIT MASK FOR VSIU
                                                                                        ;TRAPS
047506  012704  000034                             MOV     #S0M0M1,R4              ;SET BIT MASKS TO CHECK
047512  012705  000054                             MOV     #S1M0M1,R5              ;GROUP 1 FIRST
047516  050204                             VSCMA:  BIS     R2,R4
047520  050205                                     BIS     R2,R5
047522  010237  177746                             MOV     R2,@#CONTRL
047526  032737  010000  177746     1$:             BIT     #VCIP,@#CONTRL
047534  001374                                     BNE     1$

047536  010437  177746                             MOV     R4,@#CONTRL             ;MAKE 'NOP'' LIST IN GROUP
047542  005737  051572                             TST     @#2$+2000               ;BEING TESTED
047546  010537  177746                             MOV     R5,@#CONTRL
047552  005737  047572                             TST     @#2$
047556  042737  000014  177746                     BIC     #M0M1,@#CONTRL
047564  052737  002001  177746                     BIS     #FVPE+DT,@#CONTRL       ;FORCE VALID STORE PARITY ERROR

047572  000240                             2$:     NOP                             ;REFERENCE OF THIS INSTRUCTION
                                                                                   ;WILL FORCE A VALID STORE
                                                                                   ;PARITY ERROR

047574  032737  000010  177752                     BIT     #10,@#HITMIS            ;CHECK THAT THE REFERENCE
047602  001407                                     BEQ     3$                      ;WHICH CAUSED THE V-STORE
047604  013737  177746  001556                     MOV     @#CONTRL,$REG0          ;PARITY ERROR WAS A MISS
047612  013737  177744  001560                     MOV     @#MSER,$REG1            ;TEXT-DATA REFERENCE WHICH
047620  104104                                     ERROR   104                     ;CAUSED A PARITY ERROR
                                                                                   ;(IN THE VALID STORE)
                                                                                   ;SHOULD HAVE BEEEN A MISS-
                                                                                   ;IT WAS NOT.

047622  032737  100000  177746             3$:     BIT     #VSPE,@#CONTRL          ;DID VALID STORE PARITY ERROR
                                                                                    ;SET?
047630  001004                                     BNE     4$
047632  013737  177746  001556                     MOV     @#CONTRL,$REG0
047640  104106                                     ERROR   106                     ;VALID STORE PARITY ERROR BIT
                                                                                   ;DID NOT SET IN CCR WHEN
                                                                                   ;PARITY ERROR FROM (V-STORE)
                                                                                   ;WAS FORCED

047642  052737  100000  177746     4$:             BIS     #VSPE,@#CONTRL          ;CLEAR VSPE
047650  032737  100000  177746                     BIT     #VSPE,@#CONTRL          ;CHECK
047656  001404                                     BEQ     5$
047660  013737  177746  001556                     MOV     @#CONTRL,$REG0
047666  104112                                     ERROR   112                     ;VALID STORE PARITY ERROR
                                                                                   ;BIT COULD NOT BE CLEARED IN CCR

047670  013737  177744  177744             5$:     MOV     @#MSER,@#MSER           ;CLEAR MEMORY SYSTEM ERROR
                                                                                    ;REGISTER
```

```
047676  012700  020000                  MOV     #VSIU,R0
047702  074002                          XOR     R0,R2                           ;DONE VALID STORE B?
047704  001304                          BNE     VSCMA                           ;GO CHECK V-STORE B
047706  032705  000020                  BIT     #S0,R5                          ;CHECKED GROUP 0
047712  001007                          BNE     6$

047714  012704  000054                  MOV     #S1M0M1,R4                      ;SET UP BIT MASKS ITS CHECK
047720  012705  000034                  MOV     #S0M0M1,R5                      ;GROUP 0
047724  005002                          CLR     R2
047726  000137  047516                  JMP     VSCMA

047732  005037  177746          6$:     CLR     @#CONTRL
047736                          KT:



                                ;;**************************************************************
                                ;*TEST 55        CHECK BYP ON KERNEL PAGE BITS
                                        ;THIS TEST IS EXECUTED ONLY ON KB11-E,KB11-EM,AND MODIFIED KB11-B/C (KB11CM)
                                ;;**************************************************************
047736  000004                  TST55:  SCOPE
047740  012737  000012  001676          MOV     #12,$TIMES      ;;DO 12 ITERATIONS
047746  105737  001714                  TSTB    KB11CM
047752  001003                          BNE     5$              ;BR IF MOIFIED 11/70 (KB11CM)
047754  005737  001712                  TST     KB11E           ;IS IT A KB11-E OR KB11-EM?
047760  001444                          BEQ     TST56           ;;
047762  012700  172300          5$:     MOV     #KIPDR0,R0      ;POINT TO KIPDR0
047766  005010                  4$:     CLR     (R0)            ;CLEAR KIPDR
047770  032710  100000                  BIT     #BYP,(R0)       ;DID BYP CLEAR?
047774  001405                          BEQ     1$              ;BRANCH IF YES
047776  010037  001556                  MOV     R0,$REG0
050002  011037  001560                  MOV     (R0),$REG1
050006  104123                          ERROR   123             ;BYP STUCK SET

050010  052710  100000          1$:     BIS     #BYP,(R0)               ;SET BYP
050014  032710  100000                  BIT     #BYP,(R0)       ;IS IT SET?
050020  001005                          BNE     2$              ;BRANCH IF YES
050022  010037  001556                  MOV     R0,$REG0
050026  011037  001560                  MOV     (R0),$REG1
050032  104124                          ERROR   124             ;BYP STUCK CLEAR

050034  042710  100000          2$:     BIC     #BYP,(R0)       ;CLEAR BYP
050040  032710  100000                  BIT     #BYP,(R0)       ;IS IT CLEAR?
050044  001405                          BEQ     3$              ;BRANCH IF YES
050046  010037  001556                  MOV     R0,$REG0
050052  011037  001560                  MOV     (R0),$REG1
050056  104123                          ERROR   123             ;BYP STUCK SET

050060  062700  000002          3$:     ADD     #2,R0           ;POINT TO NEXT PDR
050064  020027  172340                  CMP     R0,#KDPDR7+2    ;ARE WE FINISHED?
050070  001336                          BNE     4$              ;BRANCH IF NOT

                                ;;**************************************************************
                                ;*TEST 56        CHECK BYP ON SUPERVISOR PAGE BITS
                                        ;*THIS TEST IS EXECUTED ONLY ON KB11-E, KB11-EM, AND MODIFIED KB11-B/C (KB11CM).
                                ;;**************************************************************
050072  000004                  TST56:  SCOPE
```

```
050074  012737  000012  001676          MOV     #12,$TIMES      ;;DO 12 ITERATIONS
050102  105737  001714                  TSTB    KB11CM
050106  001003                          BNE     5$              ;BR IF MOIFIED 11/70 (KB11CM)
050110  005737  001712                  TST     KB11E           ;IS IT A KB11-E OR KB11-EM?
050114  001444                          BEQ     TST57           ;;
050116  012700  172200          5$:     MOV     #SIPDR0,R0      ;POINT TO SIPDR0
050122  005010                  4$:     CLR     (R0)            ;CLEAR SIPDR
050124  032710  100000                  BIT     #BYP,(R0)       ;DID BYP CLEAR?
050130  001405                          BEQ     1$              ;BRANCH IF YES
050132  010037  001556                  MOV     R0,$REG0
050136  011037  001560                  MOV     (R0),$REG1
050142  104130                          ERROR   130             ;BYP STUCK SET

050144  052710  100000          1$:     BIS     #BYP,(R0)       ;SET BYP
050150  032710  100000                  BIT     #BYP,(R0)       ;DID IT SET?
050154  001005                          BNE     2$              ;BRANCH IF YES
050156  010037  001556                  MOV     R0,$REG0
050162  011037  001560                  MOV     (R0),$REG1
050166  104131                          ERROR   131             ;BYP STUCK CLEAR

050170  042710  100000          2$:     BIC     #BYP,(R0)       ;CLEAR BYP
050174  032710  100000                  BIT     #BYP,(R0)       ;IS IT CLEAR?
050200  001405                          BEQ     3$              ;BRANCH IF YES
050202  010037  001556                  MOV     R0,$REG0
050206  011037  001560                  MOV     (R0),$REG1
050212  104130                          ERROR   130             ;BYP STUCK SET

050214  062700  000002          3$:     ADD     #2,R0           ;POINT TO NEXT PDR
050220  020027  172240                  CMP     R0,#SDPDR7+2    ;ARE WE FINISHED?
050224  001336                          BNE     4$              ;BRANCH IF NO
                                ;;*****************************************************************
                                ;*TEST 57               CHECK BYP ON USER PAGE BITS
                                        ;*THIS TEST IS EXECUTED ONLY ON KB11-E, KB11-EM, AND MODIFIED KB11-B/C (KB11CM).
                                ;;*****************************************************************
050226  000004                  TST57:  SCOPE
050230  012737  000012  001676          MOV     #12,$TIMES      ;;DO 12 ITERATIONS
050236  105737  001714                  TSTB    KB11CM
050242  001003                          BNE     5$              ;BR IF MOIFIED 11/70 (KB11CM)
050244  005737  001712                  TST     KB11E           ;IS IT A KB11-E OR KB11-EM?
050250  001444                          BEQ     TST60           ;;
050252  012700  177600          5$:     MOV     #UIPDR0,R0      ;POINT TO UIPDR0
050256  005010                  4$:     CLR     (R0)            ;CLEAR UIPDR
050260  032710  100000                  BIT     #BYP,(R0)       ;DID BYP CLEAR?
050264  001405                          BEQ     1$              ;BRANCH IF YES
050266  010037  001556                  MOV     R0,$REG0
050272  011037  001560                  MOV     (R0),$REG1
050276  104132                          ERROR   132             ;BYP STUCK SET

050300  052710  100000          1$:     BIS     #BYP,(R0)       ;SET BYP
050304  032710  100000                  BIT     #BYP,(R0)       ;IS IT SET?
050310  001005                          BNE     2$              ;BRANCH IF YES
050312  010037  001556                  MOV     R0,$REG0
050316  011037  001560                  MOV     (R0),$REG1
050322  104133                          ERROR   133             ;BYP STUCK CLEAR

050324  042710  100000          2$:     BIC     #BYP,(R0)       ;CLEAR BYP
```

```
 9034  050330  032710  100000                          BIT     #BYP,(R0)          ;IS IT CLEAR?
 9035  050334  001405                                  BEQ     3$                 ;BRANCH IF YES
 9036  050336  010037  001556                          MOV     R0,$REG0
 9037  050342  011037  001560                          MOV     (R0),$REG1
 9038  050346  104132                                  ERROR   132                ;BYP STUCK SET
 9039
 9040  050350  062700  000002                  3$:     ADD     #2,R0              ;POINT TO NEXT PDR
 9041  050354  020027  177640                          CMP     R0,#UDPDR7+2       ;ARE WE FINISHED?
 9042  050360  001336                                  BNE     4$                 ;BRANCH IF NO
 9043                                          ;;*****************************************************************
 9044                                          ;*TEST 60          CHECK CACHE BYPASS ON VIRTUAL PAGE
 9045                                                  ;*THIS TEST IS EXECUTED ONLY ON KB11-EM AND 11/74          (KB11CM)
 9046                                          ;;*****************************************************************
 9047  050362  000004                          TST60:  SCOPE
 9048
 9049
 9050  050364  013746  177776                          MOV     @#PS,-(SP)         ;CLEAR T BIT IF SET
 9051  050370  042716  000020                          BIC     #20,(SP)
 9052  050374  012746  050402                          MOV     #1$,-(SP)
 9053  050400  000002                                  RTI
 9054  050402                                  1$:
 9055
 9056  050402  105737  001714                          TSTB    KB11CM
 9057  050406  001005                                  BNE     VPBP
 9058  050410  105737  001713                          TSTB    KB11EM             ;IS IT A KB11-EM?
 9059  050414  001002                                  BNE     VPBP               ;BR IF YES
 9060  050416  000137  051214                          JMP     VPBPE
 9061  050422  012704  100000                  VPBP:   MOV     #100000,R4         ;INITIALIZE APF, PAGE PAR = 4
 9062  050426  012705  172350                          MOV     #KIPAR4,R5
 9063  050432  012703  172310                          MOV     #KIPDR4,R3
 9064  050436  005037  177746                  VPBPA:  CLR     @#CONTRL
 9065  050442  032737  010000  177746          1$:     BIT     #VCIP,@#CONTRL     ;WAIT FOR VCIP TO CLEAR
 9066  050450  001374                                  BNE     1$
 9067
 9068  050452  012700  050422                          MOV     #VPBP,R0           ;MAKE TEST CODE HIT IN GROUP 1
 9069  050456  012701  001000                          MOV     #1000,R1
 9070  050462  012737  000034  177746          2$:     MOV     #S0M0M1,@#CONTRL
 9071  050470  005760  002000                          TST     2000(R0)
 9072  050474  012737  000054  177746                  MOV     #S1M0M1,@#CONTRL
 9073  050502  005720                                  TST     (R0)+
 9074  050504  077112                                  SOB     R1,2$
 9075
 9076  050506  042737  000014  177746                  BIC     #M0M1,@#CONTRL
 9077  050514  012700  062764                          MOV     #TSTDT1,R0         ;MAKE TEST-DATA HIT IN
 9078  050520  012701  001000                          MOV     #1000,R1           ;GROUP 0
 9079  050524  012737  000020  177746                  MOV     #S0,@#CONTRL
 9080  050532  005720                          9$:     TST     (R0)+
 9081  050534  077102                                  SOB     R1,9$
 9082
 9083  050536  012737  000040  177746                  MOV     #S1,@#CONTRL       ;FORCE REPLACE GROUP 1
 9084
 9085  050544  005037  172340                          CLR     @#KIPAR0           ;MAP 0-4K VIRTUAL INTO
 9086  050550  012737  077406  172300                  MOV     #77406,@#KIPDR0    ;0-4K PHYSICAL (TEST PROGRAM)
 9087  050556  012737  000200  172342                  MOV     #200,@#KIPAR1      ;MAP 4-8K VIRTUAL INTO
 9088  050564  012737  077406  172302                  MOV     #77406,@#KIPDR1    ;4-8K PHYSICAL (TEST PROGRAM)
 9089  050572  012737  000400  172344                  MOV     #400,@#KIPAR2      ;MAP 8-12K VIRTUAL TO
```

```
050600  012737  077406  172304          MOV     #77406,@#KIPDR2 ;8-12K PHYSICAL
050606  012737  000600  172346          MOV     #600,@#KIPAR3   ;MAP 12-16K VIRTUAL TO
050614  012737  077406  172306          MOV     #77406,@#KIPDR3 ;12-16K PHYSICAL
050622  012737  177600  172356          MOV     #177600,@#KIPAR7;MAP I/O PAGE THROUGH PAGE7
050630  012737  077406  172316          MOV     #77406,@#KIPDR7

                                                                ;SET UP PAR,PDR REGISTERS TO
                                                                ;MAP THE TEST DATA BUFFER THRU THE
                                                                ;VITUAL PAGE BEING TESTED
050636  012702  062764                  MOV     #TSTDT1,R2      ;PHYSICAL ADDRESS
050642  010200                          MOV     R2,R0           ;COPY IT
050644  072027  177772                  ASH     #-6,R0          ;FORM THE PAF (BLOCK #)
050650  010015                          MOV     R0,(R5)         ;SET UP PAF INSIDE THE PAR
050652  012713  010406                  MOV     #10406,(R3)     ;SET UP PDR

                                                                ;FORM THE VIRTUAL ADDRESS FOR
                                                                ;THE TEST DATA BUFFER
050656  042702  177700                  BIC     #177700,R2      ;CLEAR APF BIT POSITIONS
050662  050402                          BIS     R4,R2           ;SET APF BITS TO POINT TO THE
050664  010200                          MOV     R2,R0           ;PAR FOR THE VIRTUAL PAGE BEING
                                                                ;TESTED
                                                                ;R2 CONTAINS THE VIRTUAL ADDRESS
                                                                ;OF THE TEST DATA BUFFER
050666  012737  000020  172516          MOV     #20,@#MMR3      ;ENABLE KT - 22 BIT MODE
050674  012737  000001  177572          MOV     #1,@#MMR0

050702  012701  001000                  MOV     #1000,R1        ;COUNT
050706  005712                  3$:     TST     (R2)
050710  032737  000010  177752          BIT     #10,@#HITMIS    ;HIT?
050716  001021                          BNE     4$              ;YES
050720  013737  177746  001556          MOV     @#CONTRL,$REG0
050726  010537  001560                  MOV     R5,$REG1        ;PAR ADDRESS
050732  011537  001562                  MOV     (R5),$REG2      ;PAR CONTENTS
050736  011337  001564                  MOV     (R3),$REG3      ;PDR CONTENTS
050742  010237  001566                  MOV     R2,$REG4        ;TEST DATA ADDRESS (VA)
050746  005037  177572                  CLR     @#MMR0          ;TURN OFF MEM MAN
050752  104125                          ERROR   125             ;TEST-DATA-BUFFER WAS REFERENCED,
                                                                ;IT WAS FOUND TO BE A MISS.
                                                                ;SHOULD HAVE BEEN A HIT
                                                                ;BECAUSE IT WAS MADE HIT
                                                                ;IN GROUP 0 BEFORE REFERENCING
050754  012737  000001  177572          MOV     #1,@#MMR0               ;TURN MM BACK ON
050762  062702  000002          4$:     ADD     #2,R2
050766  077131                          SOB     R1,3$
050770  010002                          MOV     R0,R2           ;COPY VIRTUAL ADDRESS OF TEST-DATA BUFFER
050772  052713  100000                  BIS     #BYP,(R3)       ;SET BYPASS IN PDR
050776  012701  001000                  MOV     #1000,R1        ;NOW REFERENCE THE TEST LOCATIONS
                                                                ;THAT WERE MADE HITS PREVIOUSLY
051002  005710                  5$:     TST     (R0)            ;CHECK THEY ARE BEING BYPASSED
051004  032737  000010  177752          BIT     #10,@#HITMIS    ;MISS?
051012  001421                          BEQ     6$              ;YES
051014  013737  177746  001556          MOV     @#CONTRL,$REG0
051022  010537  001560                  MOV     R5,$REG1        ;PAR ADDRESS
051026  011537  001562                  MOV     (R5),$REG2      ;PAR CONTENTS
051032  011337  001564                  MOV     (R3),$REG3      ;PDR CONTENTS
051036  010037  001566                  MOV     R0,$REG4        ;TEST DATA ADDRESS (VA)
```

```
051042  005037  177572                          CLR     @#MMR0          ;TURN OFF MM
051046  104126                                  ERROR   126             ;TEST DATA WAS NOT A MISS WHEN
                                                                        ;IT WAS REFERENCED WITH CACHE
                                                                        ;BYPASS (ON VIRTUAL PAGE) SET.
                                                                        ;TEST DATA WAS PREVIOUSLY MADE HIT
                                                                        ;IN GROUP 0. IT WAS MAPPED
                                                                        ;THROUGH A PAR,PDR SET
                                                                        ;(BEING TESTED). BYPASS BIT WAS
                                                                        ;SET IN THE PDR AND TEST-LOC
                                                                        ;WAS REFERENCED.  IT SHOULD HAVE
                                                                        ;BEEN A MISS (BECAUSE OF BYPASS)
                                                                        ;PROBABLE FAULT: SETTING CACHE
                                                                        ;BYPASS IN PDR DOES NOT BYPASS
                                                                        ;VIRTUAL REFERENCES MAPPED THRU
                                                                        ;THAT PAGE.
051050  012737  000001  177572                  MOV     #1,@#MMR0       ;TURN MM BACK ON


051056  062700  000002                  6$:     ADD     #2,R0
051062  077131                                  SOB     R1,5$
051064  042713  100000                          BIC     #BYP,(R3)       ;CLEAR BYPASS IN PDR
051070  012701  001000                          MOV     #1000,R1        ;REFERENCE THE TEST-DATA AND
                                                                        ;MAKE SURE IT WAS INVALIDATED
                                                                        ;ON PREVIOUS BYPASS
051074  005712                          7$:     TST     (R2)            ;REFERENCE TEST DATA
051076  032737  000010  177746                  BIT     #10,@#CONTRL    ;MISS?
051104  001421                                  BEQ     8$
051106  013737  177746  001556                  MOV     @#CONTRL,$REG0
051114  010537  001560                          MOV     R5,$REG1        ;PAR ADDRESS
051120  011537  001562                          MOV     (R5),$REG2      ;PAR CONTENTS
051124  011337  001564                          MOV     (R3),$REG3      ;PDR CONTENTS
051130  010237  001566                          MOV     R2,$REG4        ;TEST DATA ADDRESS (VIRTUAL ADDRESS)
051134  005037  177572                          CLR     @#MMR0          ;TURN OFF MM
051140  104127                                  ERROR   127             ;TEST-DATA REFERENCE WAS NOT
                                                                        ;A MISS.  PROBABLE FAILURE:
                                                                        ;PREVIOUSLY CACHED TEST DATA
                                                                        ;LOCATIONS WERE NOT INVALIDATED
                                                                        ;(IN THE CACHE) WHEN CACHE
                                                                        ;BYPASS WAS FORCED ON REFERENCES THROUGH
                                                                        ;THE VIRTUAL PAGE (BEING TESTED).
051142  012737  000001  177572                  MOV     #1,@#MMR0       ;TURN MM BACK ON

051150  062702  000002                  8$:     ADD     #2,R2           ;NEXT LOCATION
051154  077131                                  SOB     R1,7$           ;DONE?


051156  005037  177572                          CLR     @#MMR0          ;DISABLE KT
051162  005037  172516                          CLR     @#MMR3
051166  062704  020000                          ADD     #20000,R4       ;INITIALIZE APF FIELD MASK FOR THE
051172  062705  000002                          ADD     #2,R5           ;NEXT PAR TO BE TESTED
051176  062703  000002                          ADD     #2,R3           ;NEXT PDR TO BE TESTED

051202  020327  172316                          CMP     R3,#KIPDR0+16   ;DONE TESTING EVERY PDR?
051206  001402                                  BEQ     VPBPE
051210  000137  050436                          JMP     VPBPA
051214                                  VPBPE:
```

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

                                        .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

051214                                  $EOP:
051214  000004                                  SCOPE
051216  005037  001502                          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
051222  005037  001676                          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
051226  005237  001500                          INC     $PASS           ;;INCREMENT THE PASS NUMBER
051232  042737  100000  001500                  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
051240  005327                                  DEC     (PC)+           ;;LOOP?
051242  000001                          $EOPCT: .WORD   1
051244  003031                                  BGT     $DOAGN          ;;YES
051246  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
051250  000001                          $ENDCT: .WORD   1
051252  051242                                  $EOPCT
051254  104400  051334                          TYPE    ,$ENDMG         ;;TYPE ''END PASS #''
051260  013746  001500                          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
051264  104410                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
051266  104400  051351                          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
051272  013700  000042                  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
051276  001414                                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
051300  012703  125252                          MOV     #125252,R3
051304  004737  054544                          JSR     PC,CHAINQ
051310  013700  000042                          MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
051314  001405                                  BEQ     $DOAGN          ;;RETURN ADDRESS
051316  000005                                  RESET                   ;;CLEAR THE WORLD
051320  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
051322  000240                                  NOP                     ;;SAVE ROOM
051324  000240                                  NOP                     ;;FOR
051326  000240                                  NOP                     ;;ACT11
051330                                  $DOAGN:
051330  000137  005144                          JMP     @#LOOP          ;;RETURN
051334  005015  047105  020104          $ENDMG: .ASCIZ  <15><12>/END PASS #/
051342  040520  051523  021440
051350     000
051351     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>)
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW14=1         LOOP ON TEST
                                        ;*SW11=1         INHIBIT ITERATIONS
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*SW08=1         LOOP ON TEST IN SWR<6:0>
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT

051354                                  $SCOPE:
051354  006137  177570                          ROL     @#SWR           ;;LOOP ON PRESENT TEST?
051360  100517                                  BMI     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
051362  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
051364  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
051370  012737  051410  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
051376  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
051402  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
051406  000471                                  BR      $SVLAD          ;;GO TO THE NEXT TEST
051410  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
051412  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
051416  000431                                  BR      7$              ;;LOOP ON THE PRESENT TEST
051420                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
051420  032737  000400  177570                  BIT     #BIT08,@#SWR    ;;LOOP ON SPEC. TEST?
051426  001412                                  BEQ     2$              ;;BR IF NO
051430  052737  001000  177746                  BIS     #BIT9,  @#CONTRL        ;TURN OFF CACHE
051436  013746  177570                          MOV     @#SWR,-(SP)     ;;SET DESIRED TEST NUM. FROM SWR
051442  042716  000200                          BIC     #$SWRMK,(SP)    ;;STRIP AWAY UNDESIRED BITS
051446  122637  001502                          CMPB    (SP)+,$TSTNM    ;;ON THE RIGHT TEST?
051452  001462                                  BEQ     $OVER           ;;BR IF YES
051454  105737  001505                  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
051460  001421                                  BEQ     3$              ;;BR IF NO
051462  123737  001517  001505                  CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
051470  101015                                  BHI     3$              ;;BR IF NO
051472  032737  001000  177570                  BIT     #BIT09,@#SWR    ;;LOOP ON ERROR?
051500  001404                                  BEQ     4$              ;;BR IF NO
051502  013737  001512  001510          7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
051510  000443                                  BR      $OVER
051512  105037  001505                  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
051516  005037  001676                          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
051522  000415                                  BR      1$              ;;ESCAPE TO THE NEXT TEST
051524  032737  004000  177570          3$:     BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
051532  001011                                  BNE     1$              ;;BR IF YES
051534  005737  001500                          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
051540  001406                                  BEQ     1$              ;;      INHIBIT ITERATIONS
051542  005237  001506                          INC     $ICNT           ;;INCREMENT ITERATION COUNT
051546  023737  001676  001506                  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
051554  002021                                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
051556  012737  000001  001506          1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
051564  013737  051634  001676                  MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
051572  105237  001502                  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
051576  011637  001510                          MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
051602  011637  001512                          MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
051606  005037  001700                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
051612  112737  000001  001517                  MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
051620  013737  001502  177570          $OVER:  MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER
051626  013716  001510                          MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
051632  000002                                  RTI                     ;;FIXES PS
051634  000001                          $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS


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

                                        .SBTTL  ERROR HANDLER ROUTINE

                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO ERTYPE ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1          HALT ON ERROR
                                        ;*                HALT CAN OCCUR BEFORE AND AFTER THE ERROR TYPEOUT
                                        ;*SW13=1          INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1          BELL ON ERROR
                                        ;*SW09=1          LOOP ON ERROR
                                        ;*CALL
                                        ;*       ERROR    N          ;;ERROR=EMT AND N=ERROR ITEM NUMBER

051636                                  $ERROR:
051636  105237  001505                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
051642  001775                                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
051644  013737  001502  177570                  MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
051652  005737  177570                          TST     @#SWR           ;;HALT ON ERROR = 1?
051656  100001                                  BPL     8$              ;;BRANCH IF NO
051660  000000                                  HALT                    ;;YES--HALT
051662  032737  002000  177570  8$:     BIT     #BIT10,@#SWR    ;;BELL ON ERROR?
051670  001402                                  BEQ     1$              ;;NO - SKIP
051672  104400  001702                          TYPE    ,$BELL          ;;RING BELL
051676  005237  001514          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
051702  011637  001520                          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
051706  162737  000002  001520                  SUB     #2,$ERRPC
051714  117737  127600  001516                  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
051722  032737  020000  177570                  BIT     #BIT13,@#SWR    ;;SKIP TYPEOUT IF SET
051730  001004                                  BNE     2$              ;;SKIP TYPEOUTS
051732  004737  054744                          JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
051736  104400  001707                          TYPE    ,$CRLF
051742  005737  177570          2$:     TST     @#SWR           ;;HALT ON ERROR
051746  100001                                  BPL     9$              ;;SKIP IF CONTINUE
051750  000000                                  HALT                    ;;HALT ON ERROR!
051752  022737  051320  000042  9$:     CMP     #$ENDAD,42      ;;ACT-11?
051760  001001                                  BNE     3$              ;;BRANCH IF NO
051762  000000                                  HALT                    ;;YES
051764  032737  001000  177570  3$:     BIT     #BIT09,@#SWR    ;;LOOP ON ERROR SWITCH SET?
051772  001402                                  BEQ     4$              ;;BR IF NO
051774  013716  001512                          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
052000  005737  001700          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
052004  001402                                  BEQ     5$              ;;BR IF NONE
052006  013716  001700                          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
052012                                  5$:
052012  012737  177777  177744                  MOV     #-1,@#MEMERR
052020  005037  177766                          CLR     @#CPUERR
052024  000002                                  RTI
```

```

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

                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                ;*SAVE R0-R5
                                ;*CALL:
                                ;*      SAVREG
                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                ;*
                                ;*TOP---(+16)
                                ;* +2---(+18)
                                ;* +4---R5
                                ;* +6---R4
                                ;* +8---R3
                                ;*+10---R2
                                ;*+12---R1
                                ;*+14---R0

052026                          $SAVREG:
052026  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
052030  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052032  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
052034  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
052036  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
052040  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
052042  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
052046  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
052052  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
052056  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
052062  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
052064                          $RESREG:
052064  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
052070  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
052074  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
052100  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
052104  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
052106  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
052110  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
052112  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
052114  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
052116  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
052120  000002                          RTI

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

                                .SBTTL  TYPE ROUTINE

                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
```

```
                                        ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*
                                        ;*2) USING A JSR INSTRUCTION
                                        ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
                                        ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
                                        ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE

052122  105737  001553          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
052126  100002                          BPL     1$              ;;BR IF YES
052130  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
052132  000407                          BR      3$              ;;LEAVE
052134  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
052136  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
052142  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
052144  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
052146  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
052150  012600                          MOV     (SP)+,R0        ;;RESTORE R0
052152  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
052156  000002                          RTI                     ;;RETURN
052160  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
052164  001426                          BEQ     8$
052166  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT
052172  001004                          BNE     5$
052174  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
052176  104400  001707                  TYPE    ,$CRLF
052202  000757                          BR      2$              ;;GET NEXT CHARACTER
052204  004737  052266          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
052210  123726  001552          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
052214  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
052216  013746  001550                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
052222  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
052226  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
052230  004737  052266                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
052234  105337  052332                  DECB    $CHARCNT        ;;DON'T COUNT THE NULL AS A CHARACTER
052240  000770                          BR      7$              ;;LOOP

                                        ;; ORIZONTAL TAB PROCESSOR

052242  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
052246  004737  052266          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
052252  132737  000007  052332          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
052260  001372                          BNE     9$              ;;TAB STOP
052262  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
052264  000726                          BR      2$              ;;GET NEXT CHARACTER
052266  105777  127252          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
052272  100375                          BPL     $TYPEC
052274  116677  000002  127244          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
052302  122766  000015  000002          CMPB    #CR,2(SP)       ;;BRANCH IF
```

```
052310  001003                               BNE     1$                  ;;NOT <CR>
052312  105037  052332                       CLRB    $CHARCNT            ;;
052316  000406                               BR      $TYPEX              ;;EXIT
052320  122766  000012  000002     1$:       CMPB    #LF,2(SP)           ;;BRANCH IF
052326  001402                               BEQ     $TYPEX              ;;<LF>
052330  105227                               INCB    (PC)+               ;;INC SPACE
052332  000000                     $CHARCNT:.WORD   0                   ;;COUNT
052334  000207                     $TYPEX:  RTS     PC



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

                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

052336  017646  000000             $TYPOS:  MOV     @(SP),-(SP)         ;;PICKUP THE MODE
052342  116637  000001  052561              MOVB    1(SP),$0FILL        ;;LOAD ZERO FILL SWITCH
052350  112637  052563                      MOVB    (SP)+,$OMODE+1      ;;NUMBER OF DIGITS TO TYPE
052354  062716  000002                      ADD     #2,(SP)             ;;ADJUST RETURN ADDRESS
052360  000406                              BR      $TYPON
052362  112737  000001  052561     $TYPOC:  MOVB    #1,$0FILL           ;;SET THE ZERO FILL SWITCH
052370  112737  000006  052563              MOVB    #6,$OMODE+1         ;;SET FOR SIX(6) DIGITS
052376  112737  000005  052560     $TYPON:  MOVB    #5,$OCNT            ;;SET THE ITERATION COUNT
052404  010346                              MOV     R3,-(SP)            ;;SAVE R3
052406  010446                              MOV     R4,-(SP)            ;;SAVE R4
052410  010546                              MOV     R5,-(SP)            ;;SAVE R5
052412  113704  052563                      MOVB    $OMODE+1,R4         ;;GET THE NUMBER OF DIGITS TO TYPE
052416  005404                              NEG     R4
052420  062704  000006                      ADD     #6,R4               ;;SUBTRACT IT FOR MAX. ALLOWED
052424  110437  052562                      MOVB    R4,$OMODE           ;;SAVE IT FOR USE
052430  113704  052561                      MOVB    $0FILL,R4           ;;GET THE ZERO FILL SWITCH
052434  016605  000012                      MOV     12(SP),R5           ;;PICKUP THE INPUT NUMBER
052440  005003                              CLR     R3                  ;;CLEAR THE OUTPUT WORD
052442  006105                     1$:      ROL     R5                  ;;ROTATE MSB INTO "C"
052444  000404                              BR      3$                  ;;GO DO MSB
```

```
052446  006105                    2$:       ROL     R5                  ;;FORM THIS DIGIT
052450  006105                              ROL     R5
052452  006105                              ROL     R5
052454  010503                              MOV     R5,R3
052456  006103                    3$:       ROL     R3                  ;;GET LSB OF THIS DIGIT
052460  105337  052562                      DECB    $OMODE              ;;TYPE THIS DIGIT?
052464  100016                              BPL     7$                  ;;BR IF NO
052466  042703  177770                      BIC     #177770,R3          ;;GET RID OF JUNK
052472  001002                              BNE     4$                  ;;TEST FOR 0
052474  005704                              TST     R4                  ;;SUPPRESS THIS 0?
052476  001403                              BEQ     5$                  ;;BR IF YES
052500  005204                    4$:       INC     R4                  ;;DON'T SUPPRESS ANYMORE 0'S
052502  052703  000060                      BIS     #'0,R3              ;;MAKE THIS DIGIT ASCII
052506  052703  000040            5$:       BIS     #' ,R3              ;;MAKE ASCII IF NOT ALREADY
052512  110337  052556                      MOVB    R3,8$               ;;SAVE FOR TYPING
052516  104400  052556                      TYPE    ,8$                 ;;GO TYPE THIS DIGIT
052522  105337  052560            7$:       DECB    $OCNT               ;;COUNT BY 1
052526  003347                              BGT     2$                  ;;BR IF MORE TO DO
052530  002402                              BLT     6$                  ;;BR IF DONE
052532  005204                              INC     R4                  ;;INSURE LAST DIGIT ISN'T A BLANK
052534  000744                              BR      2$                  ;;GO DO THE LAST DIGIT
052536  012605                    6$:       MOV     (SP)+,R5            ;;RESTORE R5
052540  012604                              MOV     (SP)+,R4            ;;RESTORE R4
052542  012603                              MOV     (SP)+,R3            ;;RESTORE R3
052544  016666  000002  000004              MOV     2(SP),4(SP)         ;;SET THE STACK FOR RETURNING
052552  012616                              MOV     (SP)+,(SP)
052554  000002                              RTI                         ;;RETURN
052556     000                    8$:       .BYTE   0                   ;;STORAGE FOR ASCII DIGIT
052557     000                              .BYTE   0                   ;;TERMINATOR FOR TYPE ROUTINE
052560     000                    $OCNT:    .BYTE   0                   ;;OCTAL DIGIT COUNTER
052561     000                    $OFILL:   .BYTE   0                   ;;ZERO FILL SWITCH
052562  000000                    $OMODE:   .WORD   0                   ;;NUMBER OF DIGITS TO TYPE

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

                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                  ;*REPLACED WITH SPACES.
                                  ;*CALL:
                                  ;*         MOV     NUM,-(SP)           ;;PUT THE BINARY NUMBER ON THE STACK
                                  ;*         TYPDS                       ;;GO TO THE ROUTINE

052564                            $TYPDS:
052564  010046                              MOV     R0,-(SP)            ;;PUSH R0 ON STACK
052566  010146                              MOV     R1,-(SP)            ;;PUSH R1 ON STACK
052570  010246                              MOV     R2,-(SP)            ;;PUSH R2 ON STACK
052572  010346                              MOV     R3,-(SP)            ;;PUSH R3 ON STACK
052574  010546                              MOV     R5,-(SP)            ;;PUSH R5 ON STACK
052576  012746  020200                      MOV     #20200,-(SP)        ;;SET BLANK SWITCH AND SIGN
052602  016605  000020                      MOV     20(SP),R5           ;;GET THE INPUT NUMBER
052606  100004                              BPL     1$                  ;;BR IF INPUT IS POS.
052610  005405                              NEG     R5                  ;;MAKE THE BINARY NUMBER POS.
```

```
052612  112766  000055  000001           MOVB    #'-,1(SP)          ;;MAKE THE ASCII NUMBER NEG.
052620  005000                   1$:     CLR     R0                 ;;ZERO THE CONSTANTS INDEX
052622  012703  053000                   MOV     #$DBLK,R3          ;;SETUP THE OUTPUT POINTER
052626  112723  000040                   MOVB    #' ,(R3)+          ;;SET THE FIRST CHARACTER TO A BLANK
052632  005002                   2$:     CLR     R2                 ;;CLEAR THE BCD NUMBER
052634  016001  052770                   MOV     $DTBL(R0),R1       ;;GET THE CONSTANT
052640  160105                   3$:     SUB     R1,R5              ;;FORM THIS BCD DIGIT
052642  002402                           BLT     4$                 ;;BR IF DONE
052644  005202                           INC     R2                 ;;INCREASE THE BCD DIGIT BY 1
052646  000774                           BR      3$
052650  060105                   4$:     ADD     R1,R5              ;;ADD BACK THE CONSTANT
052652  005702                           TST     R2                 ;;CHECK IF BCD DIGIT=0
052654  001002                           BNE     5$                 ;;FALL THROUGH IF 0
052656  105716                           TSTB    (SP)               ;;STILL DOING LEADING 0'S?
052660  100407                           BMI     7$                 ;;BR IF YES
052662  106316                   5$:     ASLB    (SP)               ;;MSD?
052664  103003                           BCC     6$                 ;;BR IF NO
052666  116663  000001  177777           MOVB    1(SP),-1(R3)       ;;YES--SET THE SIGN
052674  052702  000060           6$:     BIS     #'0,R2             ;;MAKE THE BCD DIGIT ASCII
052700  052702  000040           7$:     BIS     #' ,R2             ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
052704  110223                           MOVB    R2,(R3)+           ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
052706  005720                           TST     (R0)+              ;;JUST INCREMENTING
052710  020027  000010                   CMP     R0,#10             ;;CHECK THE TABLE INDEX
052714  002746                           BLT     2$                 ;;GO DO THE NEXT DIGIT
052716  003002                           BGT     8$                 ;;GO TO EXIT
052720  010502                           MOV     R5,R2              ;;GET THE LSD
052722  000764                           BR      6$                 ;;GO CHANGE TO ASCII
052724  105726                   8$:     TSTB    (SP)+              ;;WAS THE LSD THE FIRST NON-ZERO?
052726  100003                           BPL     9$                 ;;BR IF NO
052730  116663  177777  177776           MOVB    -1(SP),-2(R3)      ;;YES--SET THE SIGN FOR TYPING
052736  105013                   9$:     CLRB    (R3)               ;;SET THE TERMINATOR
052740  012605                           MOV     (SP)+,R5           ;;POP STACK INTO R5
052742  012603                           MOV     (SP)+,R3           ;;POP STACK INTO R3
052744  012602                           MOV     (SP)+,R2           ;;POP STACK INTO R2
052746  012601                           MOV     (SP)+,R1           ;;POP STACK INTO R1
052750  012600                           MOV     (SP)+,R0           ;;POP STACK INTO R0
052752  104400  053000                   TYPE    ,$DBLK             ;;NOW TYPE THE NUMBER
052756  016666  000002  000004           MOV     2(SP),4(SP)        ;;ADJUST THE STACK
052764  012616                           MOV     (SP)+,(SP)
052766  000002                           RTI                        ;;RETURN TO USER
052770  023420                   $DTBL:  10000.
052772  001750                           1000.
052774  000144                           100.
052776  000012                           10.
053000  000004                   $DBLK:  .BLKW   4

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

                                 .SBTTL  RANDOM NUMBER GENERATOR ROUTINE

                                 ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
                                 ;*WITH A RANGE OF 0 TO 2(+33)-1.
                                 ;*CALL:
                                 ;*       JSR     PC,$RAND           ;;CALL THE ROUTINE
                                 ;*       RETURN                     ;;RETURN HERE THE RANDOM
                                 ;*                                  ;;NUMBER WILL BE IN
```

```
                                  ;*                                            ;;$HINUM,$LONUM

       053010                     $RAND:
       053010  010046                     MOV     R0,-(SP)                      ;;PUSH R0 ON STACK
       053012  010146                     MOV     R1,-(SP)                      ;;PUSH R1 ON STACK
       053014  010246                     MOV     R2,-(SP)                      ;;PUSH R2 ON STACK
       053016  013700  053110             MOV     $LONUM,R0                     ;;SET R0 WITH LOW
       053022  013701  053106             MOV     $HINUM,R1                     ;;SET R1 WITH HIGH
       053026  012702  177771             MOV     #-7,R2                        ;;SET SHIFT COUNT
       053032  006300             1$:     ASL     R0                            ;;SHIFT R0 LEFT AND
       053034  006101                     ROL     R1                            ;;ROTATE CARRY INTO R1 AND
       053036  005202                     INC     R2                            ;;CHECK FOR DONE
       053040  001374                     BNE     1$                            ;;CONTINUE SHIFT LOOP
       053042  063700  053110             ADD     $LONUM,R0                     ;;ADD NUMBER TO MAKE X 129
       053046  005501                     ADC     R1                            ;;PROPOGATE CARRY
       053050  063701  053106             ADD     $HINUM,R1                     ;;ADD NUMBER TO MAKE X 129
       053054  062700  001057             ADD     #1057,R0                      ;;ADD LOW CONSTANT
       053060  005501                     ADC     R1                            ;;PROPOGATE CARRY
       053062  062701  047401             ADD     #47401,R1                     ;;ADD HIGH CONSTANT
       053066  010037  053110             MOV     R0,$LONUM                     ;;SAVE R0
       053072  010137  053106             MOV     R1,$HINUM                     ;;SAVE R1
       053076  012602                     MOV     (SP)+,R2                      ;;POP STACK INTO R2
       053100  012601                     MOV     (SP)+,R1                      ;;POP STACK INTO R1
       053102  012600                     MOV     (SP)+,R0                      ;;POP STACK INTO R0
       053104  000207                     RTS     PC                            ;;RETURN
       053106  176543             $HINUM: .WORD   176543
       053110  123456             $LONUM: .WORD   123456

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

                                  .SBTTL  TRAP DECODER

                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                  ;*GO TO THAT ROUTINE.

       053112  010046             $TRAP:  MOV     R0,-(SP)                      ;;SAVE R0
       053114  016600  000002             MOV     2(SP),R0                      ;;GET TRAP ADDRESS
       053120  005740                     TST     -(R0)                         ;;BACKUP BY 2
       053122  111000                     MOVB    (R0),R0                       ;;GET RIGHT BYTE OF TRAP
       053124  016000  053132             MOV     $TRPAD(R0),R0                 ;;INDEX TO TABLE
       053130  000200                     RTS     R0                            ;;GO TO ROUTINE


                                  .SBTTL  TRAP TABLE

                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                  ;*BY THE ''TRAP'' INSTRUCTION.

                                  ;       ROUTINE
                                  ;       -------
       053132                     $TRPAD:
       053132  052122                     $TYPE   ;;CALL=TYPE       TRAP+0(104400)  TTY TYPEOUT ROUTINE
       053134  052362                     $TYPOC  ;;CALL=TYPOC      TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
       053136  052336                     $TYPOS  ;;CALL=TYPOS      TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
```

```
053140  052376                             $TYPON  ;;CALL=TYPON     TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
053142  052564                             $TYPDS  ;;CALL=TYPDS     TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
053144  052026                             $SAVREG ;;CALL=SAVREG    TRAP+12(104412) SAVE R0-R5 ROUTINE
053146  052064                             $RESREG ;;CALL=RESREG    TRAP+14(104414) RESTORE R0-R5 ROUTINE

053150  054232                             CLEAN   ;;CALL=RSET      TRAP+16(104416) GO RESET ALL REGISTERS.
053152  054202                             ABORTT  ;;CALL=SKIPT     TRAP+20(104420) THIS WILL SKIP TO THE NEXT TEST
053154  054646                             MMDES   ;;CALL=MMSKIP    TRAP+22(104422) IF SWITCH # IS ON SKIP TO THE NEXT TEST
053156  054670                             MSIZER  ;;CALL=SIZE      TRAP+24(104424) DETERMINE THE HIGHEST ADDRESS IN MEMORY
053160  054322                             SKBADR  ;;CALL=SKPBAD    TRAP+26(104426) SKIP TEST IF ERROR ADDRESS REGISTER IS I
053162  054346                             SKBERR  ;;CALL=SKPBER    TRAP+30(104430) SKIP TEST IF ERROR REGISTER IS INOPERATI
053164  054364                             SKBCNR  ;;CALL=SKPBCN    TRAP+32(104432) SKIP TEST IF CONTROL REGISTER IS INOPERA
053166  054402                             SKBMNR  ;;CALL=SKPBMN    TRAP+34(104434) SKIP TEST IF MAINTENANCE REGISTER IS INO
053170  054420                             SKBHMR  ;;CALL=SKPBHM    TRAP+36(104436) SKIP TEST IF HIT/MISS REGISTER IS IN OPE
053172  041606                             RANDWR  ;;CALL=WRRAND    TRAP+40(104440) FILL BUFFER WITH RANDOM SEQUENCE

053174  057436                             RS4HAN  ;;CALL=CALRS4    TRAP+42(104442) DO RS04 FUNCTION
053176  056466                             RP4HAN  ;;CALL=CALRP4    TRAP+44(104444) DO RP04 FUNCTION
053200  062140                             RH4HAN  ;;CALL=CALRH4    TRAP+46(104446) DO MBT FUNCTION
053202  060372                             RK5HAN  ;;CALL=CALRK5    TRAP+50(104450) DO RK05 FUNCTION
053204  061406                             UBEHAN  ;;CALL=CALUBE    TRAP+52(104452) DO UBE FUNCTION
                                   ;;*******************************************************************************


                                   .SBTTL  POWER DOWN AND UP ROUTINES

                                   ;POWER DOWN ROUTINE
053206  012737  053334  000024     $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
053214  012737  000340  000026             MOV     #340,@#PWRVEC+2 ;;PRIO:7
053222  010046                             MOV     R0,-(SP)         ;;PUSH R0 ON STACK
053224  010146                             MOV     R1,-(SP)         ;;PUSH R1 ON STACK
053226  010246                             MOV     R2,-(SP)         ;;PUSH R2 ON STACK
053230  010346                             MOV     R3,-(SP)         ;;PUSH R3 ON STACK
053232  010446                             MOV     R4,-(SP)         ;;PUSH R4 ON STACK
053234  010546                             MOV     R5,-(SP)         ;;PUSH R5 ON STACK
053236  010637  053340                     MOV     SP,$SAVR6        ;;SAVE SP
053242  012737  053254  000024             MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
053250  000000                             HALT
053252  000776                             BR      .-2              ;;HANG UP


                                   ;POWER UP ROUTINE
053254  013706  053340             $PWRUP: MOV     $SAVR6,SP        ;;GET SP
053260  005037  053340                     CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
053264  005237  053340             1$:     INC     $SAVR6           ;;WAIT FOR THE INC
053270  001375                             BNE     1$               ;;OF  WORD
053272  012605                             MOV     (SP)+,R5         ;;POP STACK INTO R5
053274  012604                             MOV     (SP)+,R4         ;;POP STACK INTO R4
053276  012603                             MOV     (SP)+,R3         ;;POP STACK INTO R3
053300  012602                             MOV     (SP)+,R2         ;;POP STACK INTO R2
053302  012601                             MOV     (SP)+,R1         ;;POP STACK INTO R1
053304  012600                             MOV     (SP)+,R0         ;;POP STACK INTO R0
053306  012737  053206  000024             MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
053314  012737  000340  000026             MOV     #340,@#PWRVEC+2 ;;PRIO:7
053322  104400                             TYPE                     ;;REPORT THE POWER FAILURE
053324  065031                     $PWRMG: .WORD   POWERM           ;;POWER FAIL MESSAGE POINTER
053326  012716                             MOV     (PC)+,(SP)       ;;RESTART AT START
053330  004112                     $PWRAD: .WORD   START            ;;RESTART ADDRESS
```

```
053332  000002                            RTI
053334  000000                  $ILLUP:   HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
053336  000776                            BR      .-2               ;; BEFORE THE POWER DOWN WAS COMPLETE
053340  000000                  $SAVR6:   0                         ;;PUT THE SP HERE
                                ;;*****************************************************************


                                .SBTTL    ROUTINE TO SIZE MEMORY

                                ;*CALL:
                                ;*        JSR     PC,$SIZE
                                ;*        RETURN
                                ;*$LSTAD WILL CONTAIN:
                                ;*        WITH KT11 OPTION          -- LAST VIRTUAL ADDRESS OF THE LAST BANK
                                ;*        WITHOUT KT11 OPTION       -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
                                ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
                                ;*$KT11 IS THE MEMORY MANAGEMENT KEY
                                ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
                                ;*        MUST BE SETUP BEFORE THE CALL
                                ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
                                ;*        DETERMINED BY ROUTINE
                                ;* --NOTE--
                                ;*THIS ROUTINE SUPPORTS PDP 11/74.
                                ;*IF ACTUAL MEMORY IS LESS THAN THAT INDICATED BY THE SIZE REGISTER
                                ;*AND A REFERENCE IS MADE TO A MEMORY ADDRESS THAT IS GREATER THAN
                                ;*ACTUAL MEMORY BUT LESS THAN SIZE REGISTER ((INDICATED), THEN A
                                ;*MEMORY REFERENCE TIMEOUT TO VECTOR 114 WILL OCCUR.

053342  010046                  $SIZE:    MOV     R0,-(SP)          ;;SAVE R0 ON THE STACK
053344  010146                            MOV     R1,-(SP)          ;;SAVE R1 ON THE STACK
053346  010246                            MOV     R2,-(SP)          ;;SAVE R2 ON THE STACK
053350  010346                            MOV     R3,-(SP)          ;;SAVE R3 ON THE STACK
053352  013746  000004                    MOV     @#ERRVEC,-(SP)    ;;SAVE PRESENT ERROR VECTOR PS & PC
053356  013746  000006                    MOV     @#ERRVEC+2,-(SP)
053362  013746  000114                    MOV     @#114,-(SP)       ;;SAVE PRESENT PARITY VECOT PS & PC
053366  013746  000116                    MOV     @#116,-(SP)
053372  010600                            MOV     SP,R0             ;;SAVE THE STACK POINTER
053374  013737  177776  000006            MOV     @#PS,@#ERRVEC+2   ;;SET ERRVEC PS TO PRESENT PS
053402  012701  003776                    MOV     #3776,R1          ;;SETUP ADDRESS
053406  105727                            TSTB    (PC)+             ;;USE MEMORY MANAGEMENT?
053410  000200                  $KT11:    .WORD   200               ;;SET TO USE MEMORY MANAGEMENT
053412  100065                            BPL     $CORE             ;;BR IF NO
053414  012737  053560  000004            MOV     #$KTNEX,@#ERRVEC  ;;SET FOR TIMEOUT
053422  005737  177572                    TST     @#SR0             ;;KT11 ARE YOU THERE?
053426  052737  100000  053410            BIS     #100000,$KT11     ;;YES--SET KT11 KEY
053434  005046                            CLR     -(SP)             ;;INITIALIZE FOR 'PAR'' LOADING
053436  012702  172340                    MOV     #KIPAR0,R2        ;;ADDRESS OF FIRST 'PAR''
053442  012703  000010                    MOV     #^D8,R3           ;;LOAD EIGHT 'PAR.'S'' AND EIGHT 'PDR.'S''
053446  012762  077406  177740  1$:       MOV     #77406,-40(R2)    ;;PDR = 4K, UP, READ/WRITE
053454  011622                            MOV     (SP),(R2)+        ;;LOAD 'PAR''
053456  062716  000200                    ADD     #200,(SP)         ;;UPDATE FOR NEXT 'PAR''
053462  077307                            SOB     R3,1$             ;;LOOP UNTIL ALL EIGHT ARE LOADED
053464  012742  177600                    MOV     #177600,-(R2)     ;;SETUP KIPAR7 FOR I/O
053470  005042                            CLR     -(R2)             ;;SETUP KIPAR6 FOR TESTING
053472  012737  053510  000004            MOV     #2$,@#ERRVEC      ;;CATCH TIMEOUT IF NO SR3
053500  012737  000020  172516            MOV     #20,@#SR3         ;;ENABLE 22-BIT ADDRESSING
053506  000401                            BR      3$                ;;THIS PDP-11 HAS A SR3 REG.
```

```
053510  022626                   2$:       CMP     (SP)+,(SP)+          ;;CLEAN OFF THE STACK--NO SR3.
053512  005237  177572           3$:       INC     @#SR0                ;;TURN ON MEMORY MANAGEMENT
053516  012737  053550  000004             MOV     #$KTOUT,@#ERRVEC     ;;SET FOR TIME OUT
053524  012737  053672  000114             MOV     #$MTMOUT,@#114       ;;SET FOR MEM REF TIMEOUT
053532  005737  143776           4$:       TST     @#143776             ;;TRAP ON NON-EX-MEM
053536  062712  000040                     ADD     #40,(R2)             ;;MAKE A 1K STEP
053542  023712  172356                     CMP     @#KIPAR7,(R2)        ;;LAST ONE?
053546  101371                             BHI     4$                   ;;NO--TRY IT
053550  011202                   $KTOUT:   MOV     (R2),R2              ;;GET LAST BANK+1
053552  005037  177572                     CLR     @#SR0                ;;TURN OFF MEMORY MANAGEMENT
053556  000421                             BR      $SIZEX
053560  042737  100000  053410   $KTNEX:   BIC     #100000,$KT11        ;;KT11 NON-EXISTENT
053566  012737  053616  000004   $CORE:    MOV     #$CROUT,@#ERRVEC     ;;SET FOR TIMEOUT
053574  005002                             CLR     R2                   ;;SET UP BANK
053576  062701  004000           1$:       ADD     #4000,R1             ;;INCREMENT BY 1K
053602  062702  000040                     ADD     #40,R2               ;;1K STEP
053606  005711                             TST     (R1)                 ;;TRAP ON TIME OUT
053610  022701  177776                     CMP     #177776,R1           ;;LAST ONE
053614  001370                             BNE     1$                   ;;NO--TRY AGAIN
053616  162701  004000           $CROUT:   SUB     #4000,R1
053622  162702  000040           $SIZEX:   SUB     #40,R2               ;;DROP BACK
053626  010006                             MOV     R0,SP                ;;RESTORE THE STACK
053630  012637  000116                     MOV     (SP)+,@#116          ;;RESTOR PARITY VECTOR
053634  012637  000114                     MOV     (SP)+,@#114
053640  012637  000006                     MOV     (SP)+,@#ERRVEC+2     ;;RESTORE ERROR VECTOR
053644  012637  000004                     MOV     (SP)+,@#ERRVEC
053650  010137  053724                     MOV     R1,$LSTAD            ;;LAST ADDRESS
053654  010237  053726                     MOV     R2,$LSTBK            ;;LAST BANK
053660  012603                             MOV     (SP)+,R3             ;;RESTORE R3
053662  012602                             MOV     (SP)+,R2             ;;RESTORE R2
053664  012601                             MOV     (SP)+,R1             ;;RESTORE R1
053666  012600                             MOV     (SP)+,R0             ;;RESTORE R0
053670  000207                             RTS     PC
053672  032737  000001  177744   $MTMOUT:  BIT     #BIT0,@#MEMERR       ;;MAKE SURE TRAP TO 114 IS DUE
053700  001005                             BNE     1$                   ;;TO MEMORY REFERENCE TIMEOUT
                                                                        ;;IF NOT, IS IT AN ABORT?.
053702  032737  100000  177744             BIT     #BIT15,@#MEMERR      ;;CPU ABORT?
053710  001001                             BNE     1$                   ;;IF YES, EXIT OUT
053712  000002                             RTI                          ;;IF NOT, CONTINUE
053714  012737  177777  177744   1$:       MOV     #-1,@#MEMERR         ;;CLEAR THE MEM ERROR REG
053722  000712                             BR      $KTOUT
053724  000000                   $LSTAD:   .WORD   0                    ;;CONTAINS THE LAST ADDRESS
053726  000000                   $LSTBK:   .WORD   0                    ;;CONTAINS THE LAST BANK

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

                                 .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


```

```
053730  104412                    $DB20:  SAVREG                          ;;SAVE ALL REGISTERS
053732  016601  000002                    MOV     2(SP),R1                ;;PICKUP THE POINTER TO LOW WORD
053736  012705  054047                    MOV     #$OCTVL+13.,R5          ;;POINTER TO DATA TABLE
053742  012704  000014                    MOV     #12.,R4                 ;;DO ELEVEN CHARACTERS
053746  012703  177770                    MOV     #^C7,R3                 ;;MASK
053752  012100                            MOV     (R1)+,R0                ;;LOWER WORD
053754  012101                            MOV     (R1)+,R1                ;;HIGH WORD
053756  005002                            CLR     R2                      ;;TERMINATOR
053760  110245                    1$:     MOVB    R2,-(R5)                ;;PUT CHARACTER IN DATA TABLE
053762  010002                            MOV     R0,R2                   ;;GET THIS DIGIT
053764  005304                            DEC     R4                      ;;COUNT THIS CHARACTER
053766  003007                            BGT     3$                      ;;BR IF NOT THE LAST DIGIT
053770  001405                            BEQ     2$                      ;;BR IF IT IS THE LAST DIGIT
053772  005205                            INC     R5                      ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
053774  010566  000002                    MOV     R5,2(SP)                ;;ASCIZ CHAR. & PUT IT ON THE STACK
054000  104414                            RESREG                          ;;RESTORE ALL REGISTERS
054002  000207                            RTS     PC                      ;;RETURN TO USER
054004  006203                    2$:     ASR     R3                      ;;POSITION THE MASK FOR THE LAST DIGIT
054006  006001                    3$:     ROR     R1                      ;;POSITION THE BINARY NUMBER FOR
054010  006000                            ROR     R0                      ;;       THE NEXT OCTAL DIGIT
054012  006001                            ROR     R1
054014  006000                            ROR     R0
054016  006001                            ROR     R1
054020  006000                            ROR     R0
054022  040302                            BIC     R3,R2                   ;;MASK OUT ALL JUNK
054024  062702  000060                    ADD     #'0,R2                  ;;MAKE THIS CHAR. ASCII
054030  000753                            BR      1$                      ;;GO PUT IT IN THE DATA TABLE
054032  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!
054050  011637  001630            CPSPUR: MOV     (SP),$TMP1
054054  012737  054072  001632            MOV     #1$,$TMP2
054062  013737  177766  001634            MOV     @#CPUERR,$TMP3
054070  022626                            CMP     (SP)+,(SP)+     ;RESET THE STACK
054072  104150                    1$:     ERROR   150
054074  104420                            SKIPT

                                  ;THIS ROUTINE HANDLE UNEXPECTED TRAPS TO #CACHVEC.
054076  012737  054174  000114    SPUR:   MOV     #10$,@#CACHVEC
054104  013700  177744                    MOV     @#MEMERR,R0
054110  032700  000014                    BIT     #14,R0          ;SEE IF IT WAS A MAIN MEMORY ERROR.
054114  001405                            BEQ     9$
054116  013701  177740                    MOV     @#LOADRS,R1     ;IF SO THERE IS BAD PARITY IN THE
054122  042701  176000                    BIC     #176000,R1      ;CACHE AND IT MUST BE PURGED!!!
054126  005711                            TST     (R1)
054130  012737  054076  000114    9$:     MOV     #SPUR,@#CACHVEC
054136  013737  177744  001636            MOV     @#MEMERR,$TMP4  ;TRAP HERE IF AN UNEXPECTED
054144  013737  177740  001630            MOV     @#LOADRS,$TMP1  ;ERROR, PARITY, OCCURS.
054152  013737  177742  001632            MOV     @#HIADRS,$TMP2
054160  011637  001634                    MOV     (SP),$TMP3
054164  022626                            CMP     (SP)+,(SP)+
054166  104014                    1$:     ERROR   14
054170  000005                            RESET                   ;TO STOP THE ACTION OF ANY I/O DEVICE!!!!
054172  104420                            SKIPT                   ;?????
```

```
054174  022626                          10$:    CMP     (SP)+,(SP)+
054176  000137  054130                          JMP     9$

                                        ;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.
054202  011637  001630                  ABORTT: MOV     (SP),$TMP1
054206  112737  000015  001516                  MOVB    #15,$ITEMB
054214  022626                                  CMP     (SP)+,(SP)+
054216  004737  054744                          JSR     PC,ERTYPE
054222  104416                                  RSET
054224  000177  000000                          JMP     @SKAD           ;GO TO @SKAD, WHICH SHOULD
                                                                        ;BE SET TO THE
054230  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.
054232                                  CLEAN:

054232  012737  054076  000114                  MOV     #SPUR,@#CACHVEC
054240  012737  054050  000004                  MOV     #CPSPUR,@#ERRVEC
054246  011637  054320                          MOV     (SP),BACKAD
054252  012706  001500                          MOV     #STACK,SP
054256  005037  177750                          CLR     @#MAINT         ;CLEAR ALL CONTROL AND ERROR
054262  005037  177572                          CLR     @#MMR0          ;REGISTERS.
054266  005037  172516                          CLR     @#MMR3
054272  005037  177746                          CLR     @#CONTRL
054276  012737  177777  177744                  MOV     #-1,@#MEMERR
054304  005037  177766                          CLR     @#CPUERR
054310  005037  177776                          CLR     @#PSW
054314  000177  000000                          JMP     @BACKAD
054320  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
                                        ;

054322  005737  054440                  SKBADR: TST     LOAFLG
054326  001004                                  BNE     1$
054330  005737  054442                          TST     HIAFLG
054334  001001                                  BNE     1$
054336  000002                                  RTI
054340  104400                          1$:     TYPE
054342  066013                                  .WORD   ADRNG
054344  000433                                  BR      SKRNG

054346  005737  054444                  SKBERR: TST     MMRFLG
```

```
054352  001001                          BNE     1$
054354  000002                          RTI
054356  104400                  1$:     TYPE
054360  066123                          .WORD   ERRNG
054362  000424                          BR      SKRNG

054364  005737  054446          SKBCNR: TST     CONFLG
054370  001001                          BNE     1$
054372  000002                          RTI
054374  104400                  1$:     TYPE
054376  066223                          .WORD   CNRNG
054400  000415                          BR      SKRNG

054402  005737  054450          SKBMNR: TST     MANFLG
054406  001001                          BNE     1$
054410  000002                          RTI
054412  104400                  1$:     TYPE
054414  066325                          .WORD   MNRNG
054416  000406                          BR      SKRNG

054420  005737  054452          SKBHMR: TST     HIMFLG
054424  001001                          BNE     1$
054426  000002                          RTI
054430  104400                  1$:     TYPE
054432  066433                          .WORD   HMRNG

054434  022626                  SKRNG:  CMP     (SP)+,(SP)+                     ;RESET THE STACK AND GO TO THE
054436  104420                          SKIPT                                   ;NEXT TEST!!!!!

054440  000000                  LOAFLG: .WORD   0                               ;THESE ARE FLAGS USED TO DESIGNATE
054442  000000                  HIAFLG: .WORD   0                               ;EITHER A GOOD OR A BAD REGISTER.
054444  000000                  MMRFLG: .WORD   0                               ;GOOD WILL BE DESIGNATED BY A
054446  000000                  CONFLG: .WORD   0                               ;0 BAD BY A NOT ZERO!!
054450  000000                  MANFLG: .WORD   0
054452  000000                  HIMFLG: .WORD   0
054454  000000                  LOAFL2: .WORD   0
054456  000000                  HIAFL2: .WORD   0
054460  000000                  MMRFL2: .WORD   0
054462  000000                  CONFL2: .WORD   0
054464  000000                  MANFL2: .WORD   0
054466  000000                  HIMFL2: .WORD   0

                                ;THIS ROUTINE IS CALLED TO DETERMINE THE PARITY OF
                                ;A DATA PATTERN. THE PATTERN WHICH IS TAKEN BY THIS
                                ;ROUTINE AS ITS ARGUMENT SHOULD BE PUT IN R0. THEN
                                ;TRANSFER CONTROL HERE BY EXECUTING:
                                ;       JSR     PC,PARCNT
                                ;WHEN THIS ROUTINE RETURNS THE NUMBER OF ON,(1), BITS
                                ;IN R0 IS LEFT IN R2. THIS WOULD BE A NUMBER BETWEEN
                                ;0 AND 16.
054470  012701  000001          PARCNT: MOV     #1,R1
054474  005002                          CLR     R2
054476  030100                  1$:     BIT     R1,R0
054500  001401                          BEQ     2$
054502  005202                          INC     R2
054504  006301                  2$:     ASL     R1
```

```
054506  103373                           BCC     1$
054510  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.
054512  005037  177750           RESMON: CLR     @#MAINT
054516  017700  125020                   MOV     @$TKB,R0
054522  104416                           RSET
054524  005003                           CLR     R3
054526  042700  000200                   BIC     #BIT7,R0         ;GET THE CHARACTER, INITIALIZE THE REGISTERS
054532  022700  000003                   CMP     #3,R0            ;AND SEE IF THE CHARACTER WAS ^C.
054536  001032                           BNE     NOCNC            ;BRANCH AND GO TO NEXT TEST IF NOT.
054540  104400                           TYPE                     ;ECHOE THE CONTROL-C AS '^C'
054542  064766                           .WORD   CONCMS
054544  012704  002734           CHAINQ: MOV     #^D1500,R4               ;AND RESTORE THE MONITOR.
054550  012701  116710                   MOV     #BOTTOM+4,R1
054554  012702  160000                   MOV     #160000,R2
054560  012142                   1$:     MOV     (R1)+,-(R2)
054562  077402                           SOB     R4,1$
054564  012737  177777  054644           MOV     #-1,MONF         ;RESET THE MONITOR RESTORED FLAG.
054572  022703  125252                   CMP     #125252,R3
054576  001001                           BNE     STOP
054600  000207                           RTS     PC               ;IF THE MONITOR WAS RESTORED BY THE
                                                                  ;.$EOP ROUTIN RETURN TO .$EOP.
                                                                  ;OTHERWISE HALT.
054602  104400                   STOP:   TYPE                     ;TYPE THE MONITOR RESTORED MESSAGE.
054604  064772                           .WORD   MMESRS
054606  013737  054642  000060           MOV     MONTTY,@#TKVEC   ;SET THE TTY KEYBOARD INTERRUPT VECTOR
                                                                  ;TO ITS INITIAL STATE.
054614  000000                           HALT                     ;AND HALT!!
054616  012737  054512  000060           MOV     #RESMON,@#TKVEC
054624  005077  124712           NOCNC:  CLR     @$TKB            ;NOT CONTROL C SO RETURN TO NEXT TEST.
054630  152777  000100  124702           BISB    #BIT6,@$TKS
054636  000177  177366                   JMP     @SKAD            ;RETURN.
054642  000000                   MONTTY: .WORD   0        ;STORAGE FOR THE TTY KEYBOARD VECTOR'S ORIGINAL
                                                                  ;CONTENTS.
054644  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.

054646  032737  000200  177570   MMDES:  BIT     #SW7,@#SWR
054654  001001                           BNE     1$               ;IS THE SWITCH ON?
```

```
054656  000002                          RTI                             ;NO, SO RETURN.
054660  022626                  1$:     CMP     (SP)+,(SP)+
054662  104416                          RSET
054664  000177  177340                  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.
054670  010046                  MSIZER: MOV     R0,-(SP)                ;SAVE THE CONTENTS OF R0 AND R1
054672  010146                          MOV     R1,-(SP)                ;GET THE ADDRESS OF
054674  016600  000004                  MOV     4(SP),R0                ;THE CALL OF THE STACK.
054700  013710  053726                  MOV     $LSTBK,(R0)             ;GET THE ACCESSABLE BOUNDARY OF MEMORY
054704  005060  000002                  CLR     2(R0)
054710  012701  000006                  MOV     #6,R1                   ;ROTATE THE 16-BIT 'BLOCK'
                                                                        ;NUMBER 6-BITS TO THE
054714  006310                  1$:     ASL     (R0)                    ;LEFT AND TURN ON LOW ORDER
054716  006160  000002                  ROL     2(R0)                   ;BITS 1-5 LEAVING BIT-0
054722  077104                          SOB     R1,1$                   ;OFF SO AS TO CREATE
054724  052710  000076                  BIS     #76,(R0)                ;THE 22-BIT PHYSICAL ADDRESS OF
                                                                        ;THE HIGHEST WORD IN
                                                                        ;MEMORY.
054730  022020                          CMP     (R0)+,(R0)+             ;DETERMINE THE RETURN ADDRESS
054732  010066  000004                  MOV     R0,4(SP)                ;AND LEAVE ON THE STACK FOR
                                                                        ;AN RTI.
054736  012601                          MOV     (SP)+,R1                ;RESTORE R1 AND R0.
054740  012600                          MOV     (SP)+,R0
054742  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
054744  104400                  ERTYPE: TYPE
054746  001707                          .WORD   $CRLF
054750  010046                          MOV     R0,-(SP)                ;SAVE R0
054752  005000                          CLR     R0
054754  113700  001516                  MOVB    $ITEMB,R0               ;GET THE ITEM NUMBER
054760  001005                          BNE     1$                      ;ZERO?
054762  013746  001520                  MOV     $ERRPC,-(SP)            ;YES, TYPE JUST THE PC
054766  104402                          TYPOC                           ;OF THE ERROR CALL.
054770  000137  055306                  JMP     ERT5

054774  005300                  1$:     DEC     R0                      ;MAKE R0 AN INDEX FOR THE
054776  072027  000003                  ASH     #3,R0                   ;ERROR TABLE
055002  062700  001716                  ADD     #$ERRTB,R0
055006  012037  055016                  MOV     (R0)+,2$                ;TYPE EM, ERROR MESSAGE.
055012  001404                          BEQ     3$
055014  104400                          TYPE
055016  000000                  2$:     .WORD   0
055020  104400                          TYPE
```

```
055022  001707                                  .WORD   $CRLF
055024  012037  055034                  3$:     MOV     (R0)+,4$                ;TYPE DH, DATA HEADER
055030  001404                                  BEQ     5$
055032  104400                                  TYPE
055034  000000                          4$:     .WORD   0
055036  104400                                  TYPE
055040  001707                                  .WORD   $CRLF
055042  010146                          5$:     MOV     R1,-(SP)                ;SAVE R1
055044  012001                                  MOV     (R0)+,R1                ;GET DT, DATA TABLE ADDRESS
055046  001002                                  BNE     6$
055050  000137  055304                          JMP     ERT4                    ;JMP IF NO ERROR TABLE.
055054  012000                          6$:     MOV     (R0)+,R0                ;GET DF, DATA FORMAT ADDRESS
055056  105710                          ERT1:   TSTB    (R0)                    ;DATA FORMAT ENTRY EQUALS
055060  001003                                  BNE     7$                      ;ZERO?
055062  013146                                  MOV     @(R1)+,-(SP)            ;YES, SO TYPE A 16-BIT
055064  104402                                  TYPOC                           ;OCTAL NUMBER
055066  000500                                  BR      ERT2
055070  122710  000001                  7$:     CMPB    #1,(R0)                 ;FORMAT EQUALS 1?
055074  001003                                  BNE     8$
055076  013146                                  MOV     @(R1)+,-(SP)            ;YES, TYPE A DECIMAL NUMBER
055100  104410                                  TYPDS
055102  000472                                  BR      ERT2

055104  122710  000002                  8$:     CMPB    #2,(R0)                 ;FORMAT 2?
055110  001012                                  BNE     9$
055112  012146                          85$:    MOV     (R1)+,-(SP)             ;YES, TYPE A 22-BIT NUMBR
055114  004737  053730                          JSR     PC,$DB20                ;CALL $DB20 TO CONVERT THE
055120  062716  000003                          ADD     #3,(SP)                 ;BINARY TO ASCII
055124  012637  055132                          MOV     (SP)+,29$               ;TYPE THE STRING
055130  104400                                  TYPE
055132  000000                          29$:    .WORD   0
055134  000455                                  BR      ERT2

055136  122710  000004                  9$:     CMPB    #4,(R0)                 ;FORMAT 4?
055142  001004                                  BNE     10$
055144  013146                                  MOV     @(R1)+,-(SP)            ;YES, TYPE A 16-BIT
055146  104404                                  TYPOS                           ;OCTAL NUMBER SUPRESSING
055150     016                                  .BYTE   16                      ;LEADING ZEROES
055151     000                                  .BYTE   0
055152  000446                                  BR      ERT2
055154  122710  000003                  10$:    CMPB    #3,(R0)                 ;FORMAT 3?
055160  001007                                  BNE     11$
055162  013146                                  MOV     @(R1)+,-(SP)            ;YES CONVERT 16-BIT
055164  012737  177777  055312                  MOV     #-1,TVADFL              ;VIRTUAL ADDRESS TO 32-BIT
055172  004737  055320                          JSR     PC,TYPVAD               ;PHYSICAL ADDRESS AND TYPE
055176  000434                                  BR      ERT2                    ;RELOCATE ONLY IF SEG. IS ON!
055200  122710  000005                  11$:    CMPB    #5,(R0)                 ;FORMAT 5?
055204  001005                                  BNE     12$
055206  012137  055214                          MOV     (R1)+,20$               ;PRINT ASCIZ STRING
055212  104400                                  TYPE
055214  000000                          20$:    .WORD   0
055216  000426                                  BR      ERT3

055220  122710  000006                  12$:    CMPB    #6,(R0)                 ;FORMAT 6
055224  001005                                  BNE     13$
055226  005037  055312                          CLR     TVADFL
```

```
10202  055232  004737  055320                   JSR     PC,TYPVAD
10203  055236  000414                           BR      ERT2
10204
10205  055240  122710  000007          13$:     CMPB    #7,(R0)         ;FORMAT 7?
10206  055244  001010                           BNE     14$
```

```
055246  012146                                  MOV     (R1)+,-(SP)
055250  004737  053730                          JSR     PC,$DB20
055254  012637  055262                          MOV     (SP)+,45$
055260  104400                                  TYPE
055262  000000                          45$:    .WORD   0
055264  000401                                  BR      ERT2

055266  000000                          14$:    HALT                            ;?????

055270  104400                          ERT2:   TYPE                            ;PRINT A TAB AFTER TYPING AN
055272  065076                                  .WORD   $TAB                    ;ERROR TABLE ENTRY OF ALL MODES
                                                                                ;EXCEPT ASCIZ
055274  005200                          ERT3:   INC     R0                      ;POINT TO THE NEXT FORMAT BYTE
055276  005711                                  TST     (R1)                    ;IS THERE ANOTHER ENTRY?
055300  001401                                  BEQ     ERT4
055302  000665                                  BR      ERT1                    ;YES, PROCESS IT
                                                                                ;OTHERWISE:
055304  012601                          ERT4:   MOV     (SP)+,R1                ;RESTORE R1
055306  012600                          ERT5:   MOV     (SP)+,R0                ;RESTORE R0
055310  000207                                  RTS     PC                      ;AND RETURN

055312  000000                          TVADFL: .WORD   0                       ;FLAG USED TO TELL TYVAD
                                                                                ;WHETHER TO CONDITIONALLY
                                                                                ;OR UNCONDITIONALLY RELOCATE
                                                                                ;WHEN TYPING AN ADDRESS,
                                                                                ;-1 OR 0 RESPECTIVELY

055314  000000                          TVADLO: .WORD   0                       ;REGISTERS FOR THE 22-BIT
055316  000000                          TVADHI: .WORD   0                       ;ADDRESS COMPUTED BY TYVAD.

                                        ;ROUTINE WHICH CONVERTS A 16-BIT ADDRESS TO A 22-BIT
                                        ;ADDRESS. IF TVADFL IS -1, THEN CONVERT TO THE 22-BIT
                                        ;REAL ADDRESS DEPENDENT ON SEG BEING ON OR OFF FOR RELOCATION.
                                        ;IF TVADFL IS ZERO THEN UNCONDITIONAL USE THE KERNAL
                                        ;PAR WHICH IS APPROPIATE TO DO RELOCATION.
055320  104412                          TYPVAD: SAVREG
055322  016601  000002                          MOV     2(SP),R1                ;GET THE VIRTUAL
055326  010137  055314                          MOV     R1,TVADLO               ;ADDRESS
055332  005037  055316                          CLR     TVADHI
055336  005737  055312                          TST     TVADFL                  ;CONDITIONALLY RELOCATE?
055342  001404                                  BEQ     1$
055344  032737  000001  177572                  BIT     #1,@#MMR0               ;YES, SEE IF MEMORY
055352  001424                                  BEQ     2$                      ;MANAGEMENT IS ON
055354  005000                          1$:     CLR     R0                      ;RELOCATE
055356  073027  000003                          ASHC    #3,R0                   ;LEFT SHIFT R0 AND R1
055362  006300                                  ASL     R0                      ;THREE PLACES. R0 ONE
                                                                                ;MORE SO THAT IT CONTAINS
                                                                                ;2 X THE UPPER 3-BITS OF
055364  000241                                  CLC                             ;THE VIRTUAL ADDRESS
055366  006001                                  ROR     R1                      ;RESTORE R1 TO THE OFFSET
055370  006001                                  ROR     R1                      ;OF THE VIRTUAL ADDRESS
055372  006001                                  ROR     R1                      ;TO THE PAR
055374  062700  172340                          ADD     #KIPAR0,R0              ;DETERMINE THE CORRECT PAR'S
                                                                                ;ADDRESS
055400  011003                                  MOV     (R0),R3                 ;GET ITS CONTENTS
055402  005002                                  CLR     R2
```

```
055404  073227  000006                  ASHC    #6,R2             ;MAKE THE BLOCK COUNT
                                                                  ;A 22-BIT ADDRESS.
055410  060103                          ADD     R1,R3             ;ADD THE OFFSET TO THE
055412  005502                          ADC     R2                ;BASE ADDRESS

055414  010237  055316                  MOV     R2,TVADHI
055420  010337  055314                  MOV     R3,TVADLO
055424  012746  055314          2$:     MOV     #TVADLO,-(SP)     ;CALL $DB20 TO CONVERT THE
055430  004737  053730                  JSR     PC,$DB20          ;22-BIT
055434  062716  000003                  ADD     #3,(SP)           ;TYPE ONLY 8 DIGITS.
055440  012637  055446                  MOV     (SP)+,3$
055444  104400                          TYPE
055446  000000                  3$:     .WORD   0
055450  104414                          RESREG                    ;RESTORE THE REGISTERS
055452  012616                          MOV     (SP)+,(SP)        ;LEAVE ONLY THE RETURN
                                                                  ;ADDRESS ON THE STACK.
055454  000207                          RTS     PC                ;RETURN

                                        .SBTTL  SYSTEM DEVICE SIZER

                                ;THIS ROUTINE IS CALLED TO DETERMINE WHAT
                                ;CONTROLLERS AND WHAT DRIVES ARE AVAILABLE ON
                                ;THE SYSTEM.
                                ;IT USES THE FLAGS:
                                ;               RS4DFL
                                ;               RP4DFL
                                ;               RH4DFL
                                ;               RK5DFL
                                ;               UBEDFL
                                ;WHICH ARE BYTES CONTAINING A BIT FOR EACH
                                ;POSSIBLE DEVICE ON THE CONTROLLER
055456  005037  057372          SIZDEV: CLR     RS4FLG            ;INITIALIZE FLAGE
055462  005037  056422                  CLR     RP4FLG
055466  005037  062074                  CLR     RH4FLG
055472  005037  060326                  CLR     RK5FLG
055476  005037  061342                  CLR     UBEFLG
055502  005037  057374                  CLR     RS4ER1
055506  005037  056424                  CLR     RP4ER1
055512  005037  062076                  CLR     RH4ER1
055516  005037  060330                  CLR     RK5ER1
055522  005037  061344                  CLR     UBEER1
055526  104412                          SAVREG
055530  105037  056050                  CLRB    RS4DFL
055534  105037  056051                  CLRB    RP4DFL
055540  105037  056052                  CLRB    RH4DFL
055544  105037  056053                  CLRB    RK5DFL
055550  105037  056054                  CLRB    UBEDFL

055554  013737  000004  056056          MOV     @#4,SIZTM1        ;SAVE 4
055562  012737  055610  000004          MOV     #1$,@#4           ;IN CASE NON-EXISTENT REG.

055570  005777  126074                  TST     @RS4CS1           ;TEST FOR RS04
055574  004737  056070                  JSR     PC,SETREG         ;GET THE REG ADD
055600  003666                          .WORD   RS4REG
055602  004737  056130                  JSR     PC,SIZRS4         ;GET THE # OF DRIVES
055606  000403                          BR      2$
```

```

055610  022626                  1$:     CMP     (SP)+,(SP)+         ;THERE WAS NO RS04
055612  005037  177766                  CLR     @#CPUERR

055616  012737  055644  000004  2$:     MOV     #3$,@#4             ;IN CASE NON-EXISTENT REG.

055624  005777  126076                  TST     @RP4CS1             ;TEST FOR RP04
055630  004737  056070                  JSR     PC,SETREG           ;GET THE REG ADD
055634  003724                          .WORD   RP4REG
055636  004737  056214                  JSR     PC,SIZRP4           ;GET THE # OF DRIVES
055642  000403                          BR      6$

055644  022626                  3$:     CMP     (SP)+,(SP)+         ;THERE WAS NO RP04
055646  005037  177766                  CLR     @#CPUERR

055652  012737  055740  000004  6$:     MOV     #7$,@#4             ;IN CASE NON-EXISTENT REG.
055660  005777  126120                  TST     @RH4CS1             ;TEST FOR MASS BUS TESTER
055664  004737  056070                  JSR     PC,SETREG           ;GET THE REG ADD
055670  004002                          .WORD   RH4REG
055672  012777  000007  126114          MOV     #7,@RH4CS2          ;SET THE DRIVE #
055700  022777  000040  126124          CMP     #40,@RH4DT
055706  001017                          BNE     8$
055710  013737  004004  004036          MOV     RH4CS1,RH4AE
055716  062737  000074  004036          ADD     #74,RH4AE
055724  004737  056070                  JSR     PC,SETREG
055730  004034                          .WORD   RH4REX

055732  004737  056412                  JSR     PC,SIZRH4
055736  000403                          BR      8$

055740  022626                  7$:     CMP     (SP)+,(SP)+         ;THEREE WAS NO MBT
055742  005037  177766                  CLR     @#CPUERR

055746  012737  055774  000004  8$:     MOV     #9$,@#4             ;IN CASE NON-EXISTENT REG.
055754  005777  126064                  TST     @RK5DS              ;TEST FOR RK05
055760  004737  056070                  JSR     PC,SETREG           ;GET THE REG ADD
055764  004042                          .WORD   RK5REG
055766  004737  056300                  JSR     PC,SIZRK5           ;GET THE # OF DRIVES
055772  000403                          BR      10$

055774  022626                  9$:     CMP     (SP)+,(SP)+         ;THERE WAS NO RK05
055776  005037  177766                  CLR     @#CPUERR

056002  012737  056030  000004  10$:    MOV     #11$,@#4            ;IN CASE NON-EXISTENT REG
056010  005777  126050                  TST     @UBEDB              ;TEST FOR UNIBUS EXERCISER
056014  004737  056070                  JSR     PC,SETREG           ;GET THE REG ADD
056020  004062                          .WORD   UBEREG
056022  004737  056364                  JSR     PC,SIZUBE           ;GET THE #
056026  000403                          BR      12$

056030  022626                  11$:    CMP     (SP)+,(SP)+         ;THERE WAS NO UBE
056032  005037  177766                  CLR     @#CPUERR

056036  013737  056056  000004  12$:    MOV     SIZTM1,@#4          ;RESTORE 4
056044  104414                          RESREG
056046  000207                          RTS     PC
```

```

056050     000                      RS4DFL: .BYTE   0
056051     000                      RP4DFL: .BYTE   0
056052     000                      RH4DFL: .BYTE   0
056053     000                      RK5DFL: .BYTE   0
056054     000                      UBEDFL: .BYTE   0
        056056                              .EVEN

056056  000000                      SIZTM1: .WORD   0
056060  000000                      SIZTM2: .WORD   0
056062  000000                      SIZTM3: .WORD   0
056064  000000                      SIZTM4: .WORD   0
056066  000000                      SIZTM5: .WORD   0

                                    ;THIS ROUTINE IS CALED BY A:
                                    ;               JSR     PC,SETREG
                                    ;               .WORD   DEVREG
                                    ;WHERE DEVREG IS THE STARTING ADDRESS OF
                                    ;A TABLE, WHICH IS TO CONTAIN THE ADDRESS OF
                                    ;A DEVICE'S CONTROL AND STATUS REGISTERS.
                                    ;THE TABLES ARE GENERATE HERE

056070  011637  056126              SETREG: MOV     (SP),SETMP
056074  062716  000002                      ADD     #2,(SP)
056100  104412                              SAVREG
056102  017700  000020                      MOV     @SETMP,R0
056106  012001                              MOV     (R0)+,R1
056110  011002                              MOV     (R0),R2
056112  010220                      1$:     MOV     R2,(R0)+
056114  062702  000002                      ADD     #2,R2
056120  077104                              SOB     R1,1$
056122  104414                              RESREG
056124  000207                              RTS     PC

056126  000000                      SETMP:  .WORD   0


                                    ;THIS ROUTINE IS CALLED, AFTER IT HAS BEEN
                                    ;DETERMINED IF THERE IS A RS04 CONTROLLER, TO SEE
                                    ;WHT DRIVES ARE AVAILABLE.

056130  012700  000010              SIZRS4: MOV     #10,R0
056134  012701  000001                      MOV     #1,R1
056140  005002                              CLR     R2
056142  105037  056151                      CLRB    3$

056146  104442                      1$:     CALRS4
056150     001                      2$:     .BYTE   1
056151     000                      3$:     .BYTE   0               ;DO A NOP FUNCTION
056152  000000                              .WORD   0               ;FOR EACH OF POSSIBLY
056154  000000                              .WORD   0               ;8 DRIVES
056156  000000                              .WORD   0
056160  000000                              .WORD   0
056162  000000                              .WORD   0
056164  000000                              .WORD   0

```

```
056166  005737  057374                  TST     RS4ER1
056172  001001                          BNE     4$
056174  050102                          BIS     R1,R2
056176  006301                  4$:     ASL     R1
056200  105237  056151                  INCB    3$
056204  077020                          SOB     R0,1$

056206  110237  056050                  MOVB    R2,RS4DFL
056212  000207                          RTS     PC

                                ;THIS ROUTINE IS CALLED TO DETERMINE WHAT RP04
                                ;DRIVES ARE ON THE CONTROLLER

056214  012700  000010          SIZRP4: MOV     #10,R0
056220  012701  000001                  MOV     #1,R1
056224  005002                          CLR     R2
056226  105037  056235                  CLRB    3$

056232  104444                  1$:     CALRP4                          ;DO A READ IN PRESET
056234     021                  2$:     .BYTE   21                      ;FOR EACH OF UP TO
056235     000                  3$:     .BYTE   0                       ;8 DRIVES.
056236  000000                          .WORD   0
056240  000000                          .WORD   0
056242  000000                          .WORD   0
056244  000000                          .WORD   0
056246  000000                          .WORD   0
056250  000000                          .WORD   0

056252  005737  056424                  TST     RP4ER1
056256  001001                          BNE     4$
056260  050102                          BIS     R1,R2
056262  006301                  4$:     ASL     R1
056264  105237  056235                  INCB    3$
056270  077020                          SOB     R0,1$

056272  110237  056051                  MOVB    R2,RP4DFL
056276  000207                          RTS     PC

                                ;DETERMINE WHAT RK05 DRIVES ARE AVAILABLE.

056300  012700  000010          SIZRK5: MOV     #10,R0
056304  012701  000001                  MOV     #1,R1
056310  005002                          CLR     R2
```

```
 10474  056312  105037  056321                  CLRB    3$
 10475
```

```
056316  104450                  1$:     CALRK5                  ;DO A DRIVE RESET
056320     015                          .BYTE   15              ;FOR EACH OF 8
056321     000                  3$:     .BYTE   0               ;POSSIBLE DRIVES.
056322  000000                          .WORD   0
056324  000000                          .WORD   0
056326  000000                          .WORD   0
056330  000000                          .WORD   0
056332  000000                          .WORD   0
056334  000000                          .WORD   0

056336  005737  060330                  TST     RK5ER1
056342  001001                          BNE     4$
056344  050102                          BIS     R1,R2
056346  006301                  4$:     ASL     R1
056350  105237  056321                  INCB    3$
056354  077020                          SOB     R0,1$

056356  110237  056053                  MOVB    R2,RK5DFL
056362  000207                          RTS     PC

                                ;SET UP UBEDFL

056364  042777  000200  125500  SIZUBE: BIC     #BIT7,@UBECR1
056372  032777  000200  125472          BIT     #BIT7,@UBECR1
056400  001403                          BEQ     1$
056402  112737  000001  056054          MOVB    #1,UBEDFL
056410  000207                  1$:     RTS     PC

                                ;DETERMINE WHAT MASS BUS TESTER UNITS THERE ARE

056412  012737  000200  056052  SIZRH4: MOV     #BIT7,RH4DFL
056420  000207                          RTS     PC

                                        .SBTTL  DEVICE HANDLERS
                                ;;**************************************************************
                                ;*
                                ;*      THE FOLLOWING SIX ROUTINES:
                                ;*              RH4HAN
                                ;*              RP4HAN
                                ;*              RS4HAN
                                ;*              UBEHAN
                                ;*              RK5HAN
                                ;*      ARE O/I AND BUS TESTER DEVICE HANDLERS.
                                ;*      THEY ARE CALLED USING:
                                ;*              TRAP TABLE CALL
                                ;*      FUNCTION:.BYTE
                                ;*              UNITNUM:.BYTE
                                ;*      DISKADR1:.WORD
                                ;*      DISKADR2:.WORD
                                ;*      MEMADR1:.WORD
                                ;*      MEMADR2:.WORD
                                ;*      WORDCNT:.WORD
                                ;*      VECTOR: .WORD
                                ;*      RETURN:
                                ;*A     WHERE TRAP TABLE CALL IS ONE OF:
                                ;*              CALRH4
                                ;*              CALRP4
                                ;*              CALRS4
                                ;*              CALUBE
                                ;*              CALRK5
                                ;*B     FUNCTION IS THE PATTERN TO BE LOADED INTO THE
                                ;*      CONTROL REGISTER FUNCTION BITS, WITH EITHER
                                ;*      INTERRUPT ENABLED OR NOT.
                                ;*C     UNITNUM IS THE DRIVE NUMBER
                                ;*D     DISKADR1 AND DISKADR2 ARE THE DISK ADDRESS
                                ;*      SECTOR NUMBER
                                ;*E     MEMADR1 AND MEMADR2 ARE THE 22-BIT MEMORY
                                ;*      ADDRESS FOR THE TRANSFER.
                                ;*F     WORDCNT IS THE WORD COUNT A POSITIVE
                                ;*      NUMBER BETWEEN 0 AND 32K.
                                ;*G     VECTOR IS THE INTERRUPT HANDLER ROUTINE SPECIFIED
                                ;*      BY THE USER FOR AN INTERRUPT ENABLED FUNCTION.
                                ;*
                                ;*      WHEN THE HANDLER PROCESSES A CALL IT RETURNS
                                ;*      WITH THE FUNCTION IN PROGRESS IF THE
                                ;*      FUNCTION WAS INTERRUPT ENABLED.  WHEN THE
                                ;*      INTERRUPT OCCURS CONTROL IS GIVEN TO
                                ;*      THE USER SPECIFIED INTERRUPT HANDLER.
                                ;*      IF THE FUNCTION WAS NOT INTERRUPT
                                ;*      ENABLED THEN THE HANDLER WAITS FOR
                                ;*      FUNCTION DONE BEFORE RETURNING.
                                ;*
                                ;*      THE FLAGS:
                                ;*              XXXER1
                                ;*              XXXER2
                                ;*              XXXER3
                                ;*      WHERE XXX IS THE DEVICE, ARE USED TO
                                ;*      INDICATE AND LOG DEVICE ERRORS IN THE HANDLER.
                                ;*      XXX CAN BE RH4,RP4,RS4,UBE,RK5 OR RP3.
                                ;*      XXXER1=0        NO ERRORS
                                ;*      XXXER1=1        ERRORS WITH STATUS IN XXXER2 AND XXXER3.
                                ;*
                                ;;*************************************************************************

                                        .SBTTL          RP04 DISK HANDLER
                                ;RP04 DISK HANDLER

                                ;REGISTERS USED IN RP4HAN
056422  000000                  RP4FLG:.WORD    0
056424  000000                  RP4ER1:.WORD    0                       ;ERROR FLAGS.
056426  000000                  RP4ER2:.WORD    0
056430  000000                  RP4ER3:.WORD    0
056432  000000                  RP4ER4:.WORD    0
056434  000000                  RP4USE:.WORD    0
056436  000000                  RP4TMP:.WORD    0
056440  000000                  RP4FUN:.WORD    0
056442  000000                  RP4UNI:.WORD    0
056444  000000                  RP4DA1:.WORD    0
056446  000000                  RP4DA2:.WORD    0
056450  000000                  RP4MA1:.WORD    0
056452  000000                  RP4MA2:.WORD    0
056454  000000                  RP4WCT:.WORD    0
```

```
056456  000000                          RP4VEC:.WORD    0
056460  000000                          RP4TRK:.WORD    0
056462  000000                          RP4SEC:.WORD    0
056464  000000                          RP4CYL:.WORD    0

056466  005737  056422                  RP4HAN: TST     RP4FLG            ;SEE IF THERE IS
056472  001402                                  BEQ     RP4H1             ;ALREADY AN RP04 FUNCTION
056474  104000                                  ERROR                     ;IN PROGRESS. IF THERE
056476  000000                                  HALT                      ;IS ERROR> (SHOULD NEVER
056500  012737  000340  177776          RP4H1:  MOV     #340,@#PSW        ;HAPPEN.)
056506  011637  056436                          MOV     (SP),RP4TMP       ;RAISE THE PRIORITY
056512  062716  000016                          ADD     #16,(SP)
056516  104412                                  SAVREG                    ;GET AN ARGUMENT POINTER
056520  013700  056436                          MOV     RP4TMP,R0         ;RESET THE RETURN ADDRESS
056524  112037  056440                          MOVB    (R0)+,RP4FUN      ;FUNCTION
056530  112037  056442                          MOVB    (R0)+,RP4UNI      ;UNIT, DEVICE, NUMBER
056534  012037  056444                          MOV     (R0)+,RP4DA1      ;DISK ADDRESS
056540  012037  056446                          MOV     (R0)+,RP4DA2
056544  012037  056450                          MOV     (R0)+,RP4MA1      ;MEMORY ADDRESS
056550  012037  056452                          MOV     (R0)+,RP4MA2
056554  012037  056454                          MOV     (R0)+,RP4WCT      ;WORD COUNT
056560  012037  056456                          MOV     (R0)+,RP4VEC      ;INTERRUPT HANDLER ROUTINE
056564  005037  056424                          CLR     RP4ER1            ;CLEAR THE ERROR
056570  005037  056426                          CLR     RP4ER2            ;FLAGS
056574  005037  056430                          CLR     RP4ER3

056600  004737  057062                          JSR     PC,RP4S1          ;GO SET UP THE UNIT NUMBER
056604  004737  057132                          JSR     PC,RP4RDY         ;GET THE DEVICE READY.
056610  004737  057072                          JSR     PC,RP4S2          ;COMPUTE THE CYLINDER,
                                                                          ;TRACK AND SECTOR
056614  004737  057116                          JSR     PC,RP4S3          ;SET UP THE WORD COUNT

056620  013777  056442  125110          RP4H2:  MOV     RP4UNI,@RP4CS2    ;SET THE RP04 REGISTERS
056626  013777  056454  125074                  MOV     RP4WCT,@RP4WC     ;UP FOR THIS FUNCTION
056634  013777  056450  125070                  MOV     RP4MA1,@RP4BA
056642  013777  056452  125126                  MOV     RP4MA2,@RP4BAE
056650  013777  056446  125056                  MOV     RP4DA2,@RP4DA
056656  013777  056444  125076                  MOV     RP4DA1,@RP4DC
056664  013700  004102                          MOV     RP4V,R0           ;SET UP THE INTERRUPT
056670  012720  056742                          MOV     #RP4H4,(R0)+      ;VECTOR
056674  012710  000340                          MOV     #340,(R0)
056700  013700  056440                          MOV     RP4FUN,R0         ;LOAD THE FUNCTION
056704  010037  056422                          MOV     R0,RP4FLG         ;AND GO
056710  110077  125012                          MOVB    R0,@RP4CS1
056714  032700  000100                          BIT     #BIT6,R0          ;SEE IF THE FUNCTION
056720  001402                                  BEQ     RP4H3             ;WILL INTERRUPT WHEN
056722  104414                                  RESREG                    ;DONE. IF YES RETURN
056724  000002                                  RTI                       ;IF NOT INTERRUPTING
056726  004737  056756                  RP4H3:  JSR     PC,RP4H5          ;THEN WAIT FOR THE
056732  005037  056422                          CLR     RP4FLG            ;FUNCTION TO FINISH.
056736  104414                                  RESREG                    ;THEN RETURN.
056740  000002                                  RTI

056742  005037  056422                  RP4H4:  CLR     RP4FLG            ;WHEN THE INTERRUPT
056746  004737  056756                          JSR     PC,RP4H5          ;OCCURS CHECK FOR ERRORS
056752  000177  177500                          JMP     @RP4VEC           ;AND GO TO THE SERVICE
```

```
                                                                 ;ROUTINE.

056756  010046                          RP4H5:  MOV     R0,-(SP)
056760  053777  056442  124750          RP4H51: BIS     RP4UNI,@RP4CS2
056766  017700  124734                          MOV     @RP4CS1,R0
056772  005700                                  TST     R0               ;SEE IF THE FUNCTION
056774  100023                                  BPL     RP4H6            ;WAS COMPLETED WITHOUT
056776  032700  060000                          BIT     #60000,R0        ;ERRORS.
057002  001420                                  BEQ     RP4H6
057004  017737  124726  056426                  MOV     @RP4CS2,RP4ER2   ;IF ERRORS OCCURRED SET
057012  017737  124722  056430                  MOV     @RP4DS,RP4ER3    ;THE INDICATORS
057020  017737  124716  056432                  MOV     @RP4RR1,RP4ER4
057026  012737  177777  056424                  MOV     #-1,RP4ER1
057034  004737  057354                          JSR     PC,RP4CLR        ;CLEAR THE CONTROL
057040  012600                                  MOV     (SP)+,R0
057042  000207                                  RTS     PC
057044  105700                          RP4H6:  TSTB    R0               ;WAIT FOR READY OR
057046  100344                                  BPL     RP4H51           ;ERROR
057050  105777  124664                          TSTB    @RP4DS
057054  100341                                  BPL     RP4H51
057056  012600                                  MOV     (SP)+,R0
057060  000207                                  RTS     PC

057062  042737  177770  056442          RP4S1:  BIC     #177770,RP4UNI   ;SET UP THE DRIVE NUMBER.
057070  000207                                  RTS     PC

057072  013701  056444                  RP4S2:  MOV     RP4DA1,R1        ;COMPUTE THE DISK
057076  005000                                  CLR     R0
057100  071027  000630                          DIV     #408.,R0
057104  010137  056444                          MOV     R1,RP4DA1
057110  005037  056446                          CLR     RP4DA2
057114  000207                                  RTS     PC

057116  005437  056454                  RP4S3:  NEG     RP4WCT           ;COMPUTE VALID WORD COUNT
057122  042737  177700  056452                  BIC     #177700,RP4MA2   ;AND MEMORY ADDRESS
057130  000207                                  RTS     PC

057132  012737  000040  056434          RP4RDY: MOV     #BIT5,RP4USE     ;CLEAR CONTROLLER AND
057140  053737  056442  056434                  BIS     RP4UNI,RP4USE
057146  013777  056434  124562                  MOV     RP4USE,@RP4CS2
057154  013777  056442  124554                  MOV     RP4UNI,@RP4CS2
057162  105777  124540                  1$:     TSTB    @RP4CS1          ;DRIVES
057166  100375                                  BPL     1$
057170  013777  056442  124540                  MOV     RP4UNI,@RP4CS2
057176  012777  000021  124522                  MOV     #21,@RP4CS1      ;INITIALIZE THE DRIVE
057204  017701  124516                  2$:     MOV     @RP4CS1,R1       ;BY DOING A NOP
057210  005701                                  TST     R1               ;WAIT FOR ERROR OR
057212  100434                                  BMI     4$               ;READY
057214  105701                                  TSTB    R1
057216  100372                                  BPL     2$

057220  017700  124514                  3$:     MOV     @RP4DS,R0        ;LOOK AT THE DRIVE
                                                                 ;STATUS
057224  032700  000400                          BIT     #BIT8,R0         ;DRIVE PRESENT?
057230  001430                                  BEQ     5$
057232  032700  000100                          BIT     #BIT6,R0         ;VOLUME VALID?
```

```
057236  001425                          BEQ     5$
057240  032700  010000                  BIT     #BIT12,R0        ;ON LINE?
057244  001422                          BEQ     5$
057246  032700  040000                  BIT     #BIT14,R0        ;ANY ERRORS?
057252  001017                          BNE     5$
057254  032700  004000                  BIT     #BIT11,R0        ;WRITE LOCKED
057260  001014                          BNE     5$
057262  032700  001000                  BIT     #BIT9,  R0       ;PROGRAMMABLE DRIVE
057266  001011                          BNE     5$
057270  105700                          TSTB    R0               ;WAIT FOR DRIVE READY
057272  100344                          BPL     2$

057274  012777  010000  124456          MOV     #BIT12,@RP40F    ;SET 16-BIT MODE
057302  000207                          RTS     PC               ;RETURN READY.
057304  032701  040000          4$:     BIT     #BIT14,R1        ;ATTENTION OR ERROR?
057310  001743                          BEQ     3$
057312  005726                  5$:     TST     (SP)+
057314  017737  124416  056426          MOV     @RP4CS2,RP4ER2   ;FLAG AND RECORD
057322  017737  124412  056430          MOV     @RP4DS,RP4ER3    ;ERROR
057330  017737  124406  056432          MOV     @RP4RR1,RP4ER4
057336  012737  177777  056424          MOV     #-1,RP4ER1
057344  004737  057354                  JSR     PC,RP4CLR        ;CLR THE CONTROLLER
057350  104414                          RESREG                   ;AND DRIVES.
057352  000002                          RTI                      ;RETURN

057354  013777  056434  124354  RP4CLR: MOV     RP4USE,@RP4CS2   ;CLR THE CONTROLLER
057362  105777  124340          1$:     TSTB    @RP4CS1          ;AND DRIVES.
057366  100375                          BPL     1$
057370  000207                          RTS     PC

                                        .SBTTL          RS04 DISK HANDLE
                                ;RS04 DISK HANDLER


                                ;REGISTERS USED IN RS4HAN
057372  000000                  RS4FLG:.WORD    0
057374  000000                  RS4ER1:.WORD    0                ;ERROR FLAGS.
057376  000000                  RS4ER2:.WORD    0
057400  000000                  RS4ER3:.WORD    0
057402  000000                  RS4ER4:.WORD    0
057404  000000                  RS4USE:.WORD    0
057406  000000                  RS4TMP:.WORD    0
057410  000000                  RS4FUN:.WORD    0
057412  000000                  RS4UNI:.WORD    0
057414  000000                  RS4DA1:.WORD    0
057416  000000                  RS4DA2:.WORD    0
057420  000000                  RS4MA1:.WORD    0
057422  000000                  RS4MA2:.WORD    0
057424  000000                  RS4WCT:.WORD    0
057426  000000                  RS4VEC:.WORD    0
057430  000000                  RS4TRK:.WORD    0
057432  000000                  RS4SEC:.WORD    0
057434  000000                  RS4CYL:.WORD    0

057436  005737  057372          RS4HAN: TST     RS4FLG           ;SEE IF THERE ALREADY
057442  001402                          BEQ     RS4H1            ;IS AN RS04 FUNCTION
057444  104000                          ERROR                    ;IN PROGRESS. IF SO
```

```
057446 000000                         HALT                    ;ERROR. (SHOULD NEVER
057450 012737 000340 177776 RS4H1:    MOV     #340,@#PSW      ;HAPPEN.
057456 011637 057406                  MOV     (SP),RS4TMP
057462 062716 000016                  ADD     #16,(SP)
057466 104412                         SAVREG                  ;RAISE THE PRIORITY
057470 013700 057406                  MOV     RS4TMP,R0       ;GET A POINTER TO
057474 112037 057410                  MOVB    (R0)+,RS4FUN    ;FUNCTION
057500 112037 057412                  MOVB    (R0)+,RS4UNI    ;GET THE DRIVE NUMBER
057504 012037 057414                  MOV     (R0)+,RS4DA1    ;DISK ADDRESS
057510 012037 057416                  MOV     (R0)+,RS4DA2
057514 012037 057420                  MOV     (R0)+,RS4MA1    ;MEMORY ADDRESS
057520 012037 057422                  MOV     (R0)+,RS4MA2
057524 012037 057424                  MOV     (R0)+,RS4WCT    ;WORD COUNT
057530 012037 057426                  MOV     (R0)+,RS4VEC    ;INTERRUPT HANDLER ADDRESS
057534 005037 057374                  CLR     RS4ER1          ;CLEAR THE ERROR FLAGS
057540 005037 057376                  CLR     RS4ER2
057544 005037 057400                  CLR     RS4ER3

057550 004737 060024                  JSR     PC,RS4S1                ;SET UP UNIT (DRIVE) NUMBER
057554 004737 060116                  JSR     PC,RS4RDY       ;INITIALIZE DRIVE AND
                                                              ;CONTROLLER
057560 004737 060034                  JSR     PC,RS4S2        ;COMPUTE TRACK AND SECTOR
057564 004737 060102                  JSR     PC,RS4S3        ;COMPUTE WORD COUNT.

057570 013777 057412 124102 RS4H2:    MOV     RS4UNI,@RS4CS2  ;SET UP THE CONTROL
057576 013777 057424 124066           MOV     RS4WCT,@RS4WC   ;AND DRIVE REGISTERS
057604 013777 057420 124062           MOV     RS4MA1,@RS4BA
057612 013777 057422 124100           MOV     RS4MA2,@RS4BAE
057620 013777 057414 124050           MOV     RS4DA1,@RS4DA
057626 013700 004100                  MOV     RS4V,R0
057632 012720 057704                  MOV     #RS4H4,(R0)+    ;SET THE INTERRUPT
057636 012710 000340                  MOV     #340,(R0)
057642 013700 057410                  MOV     RS4FUN,R0
057646 010037 057372                  MOV     R0,RS4FLG
057652 110077 124012                  MOVB    R0,@RS4CS1      ;LOAD THE FUNCTION AND GO.
057656 032700 000100                  BIT     #BIT6,R0        ;SEE IF AN INTERRUPT
057662 001402                         BEQ     RS4H3           ;IS TO BE EXPECTED.
057664 104414                         RESREG                  ;IF YES THEN RETURN
057666 000002                         RTI
                                                              ;IF NOT INTERRUPTING
057670 004737 057720        RS4H3:    JSR     PC,RS4H5        ;THEN WAIT FOR THE
057674 005037 057372                  CLR     RS4FLG          ;FUNCTION TO FINISH
057700 104414                         RESREG
057702 000002                         RTI

057704 005037 057372        RS4H4:    CLR     RS4FLG          ;WHEN THE INTERRUPT OCCURS.
057710 004737 057720                  JSR     PC,RS4H5        ;MAKE SURE THERE WERE
057714 000177 177506                  JMP     @RS4VEC         ;NO ERRORS BEFORE GOING
                                                              ;TO THE INTERRUPT
                                                              ;SERVICE ROUTINE.
057720 010046               RS4H5:    MOV     R0,-(SP)
057722 053777 057412 123750 RS4H51:   BIS     RS4UNI,@RS4CS2
057730 017700 123734                  MOV     @RS4CS1,R0
057734 005700                         TST     R0              ;SEE IF THE FUNCTION
057736 100023                         BPL     RS4H6           ;WAS COMPLETED WITHOUT
057740 032700 060000                  BIT     #60000,R0       ;ERRORS
```

```
057744  001420                           BEQ     RS4H6
057746  017737  123726  057376           MOV     @RS4CS2,RS4ER2    ;IF ERRORS OCCURRED
057754  017737  123722  057400           MOV     @RS4DS,RS4ER3     ;SET THE INDICATORS
057762  017737  123716  057402           MOV     @RS4ER,RS4ER4
057770  012737  177777  057374           MOV     #-1,RS4ER1
057776  004737  060310                   JSR     PC,RS4CLR         ;THEN CLEAR THE CONTROL
060002  012600                           MOV     (SP)+,R0
060004  000207                           RTS     PC                ;AND DRIVES
060006  105700                  RS4H6:   TSTB    R0
060010  100344                           BPL     RS4H51            ;WAIT FOR READY OR
060012  105777  123664                   TSTB    @RS4DS            ;ERROR
060016  100341                           BPL     RS4H51
060020  012600                           MOV     (SP)+,R0
060022  000207                           RTS     PC

060024  042737  177770  057412  RS4S1:   BIC     #177770,RS4UNI    ;SET UP DRIVE NUMBER
060032  000207                           RTS     PC

060034  013701  057414          RS4S2:   MOV     RS4DA1,R1         ;COMPUTE A DISK
060040  005000                           CLR     R0                ;ADDRESS
060042  071027  007000                   DIV     #3584.,R0
060046  005000                           CLR     R0
060050  071027  000100                   DIV     #100,R0
060054  010037  057430                   MOV     R0,RS4TRK
060060  010137  057434                   MOV     R1,RS4CYL
060064  000300                           SWAB    R0
060066  006200                           ASR     R0
060070  006200                           ASR     R0
060072  050001                           BIS     R0,R1
060074  010137  057414                   MOV     R1,RS4DA1
060100  000207                           RTS     PC

060102  005437  057424          RS4S3:   NEG     RS4WCT            ;COMPUTE A VALID WORD
060106  042737  177700  057422           BIC     #177700,RS4MA2    ;COUNT AND MEMORY
060114  000207                           RTS     PC                ;ADDRESS
060116  012737  000040  057404  RS4RDY:  MOV     #BIT5,RS4USE      ;CLEAR CONTROLLER AND DRIVES
060124  053737  057412  057404           BIS     RS4UNI,RS4USE
060132  013777  057404  123540           MOV     RS4USE,@RS4CS2
060140  013777  057412  123532           MOV     RS4UNI,@RS4CS2
060146  105777  123516          1$:      TSTB    @RS4CS1
060152  100375                           BPL     1$
060154  013777  057412  123516           MOV     RS4UNI,@RS4CS2
060162  012777  000001  123500           MOV     #1,@RS4CS1        ;INITIALIZE THE DRIVE
060170  017701  123474          2$:      MOV     @RS4CS1,R1        ;BY DOING A NOP.
060174  005701                           TST     R1
060176  100420                           BMI     4$
060200  105701                           TSTB    R1
060202  100372                           BPL     2$

060204  017700  123472          3$:      MOV     @RS4DS,R0         ;LOOK AT THE DRIVE STATUS
060210  032700  000400                   BIT     #BIT8,R0          ;DRIVE PRESENT?
060214  001414                           BEQ     5$
060216  032700  010000                   BIT     #BIT12,R0         ;ON LINE?
060222  001411                           BEQ     5$
060224  032700  004000                   BIT     #BIT11,R0         ;WRITE LOCKED?
060230  001006                           BNE     5$
```

```
060232  105700                          TSTB    R0                  ;DRIVE READY?
060234  100355                          BPL     2$
060236  000207                          RTS     PC
060240  032701  040000          4$:     BIT     #BIT14,R1           ;ATTENTION OR ERROR?
060244  001757                          BEQ     3$
060246  005726                  5$:     TST     (SP)+
060250  017737  123424  057376          MOV     @RS4CS2,RS4ER2      ;FLAG AND RECORD THE
060256  017737  123420  057400          MOV     @RS4DS,RS4ER3       ;ERROR
060264  017737  123414  057402          MOV     @RS4ER,RS4ER4
060272  012737  177777  057374          MOV     #-1,RS4ER1
060300  004737  060310                  JSR     PC,RS4CLR           ;CLR THE CONTROLLER
060304  104414                          RESREG                      ;AND DRIVES AND RETURN.
060306  000002                          RTI

060310  013777  057404  123352  RS4CLR: MOV     RS4USE,@RS4CS1      ;CLR THE CONTROLLER
060316  105777  123346          1$:     TSTB    @RS4CS1
060322  100375                          BPL     1$
060324  000207                          RTS     PC


                                        .SBTTL          RK05 DISK HANDLER
                                ;RK05 DISK HANDLER

                                ;REGISTERS USED IN RK5HAN
060326  000000                  RK5FLG:.WORD    0
060330  000000                  RK5ER1:.WORD    0                   ;ERROR FLAGS.
060332  000000                  RK5ER2:.WORD    0
060334  000000                  RK5ER3:.WORD    0
060336  000000                  RK5ER4:.WORD    0
060340  000000                  RK5USE:.WORD    0
060342  000000                  RK5TMP:.WORD    0
060344  000000                  RK5FUN:.WORD    0
060346  000000                  RK5UNI:.WORD    0
060350  000000                  RK5DA1:.WORD    0
060352  000000                  RK5DA2:.WORD    0
060354  000000                  RK5MA1:.WORD    0
060356  000000                  RK5MA2:.WORD    0
060360  000000                  RK5WCT:.WORD    0
060362  000000                  RK5VEC:.WORD    0
060364  000000                  RK5TRK:.WORD    0
060366  000000                  RK5SEC:.WORD    0
060370  000000                  RK5CYL:.WORD    0

060372  005737  060326          RK5HAN: TST     RK5FLG              ;SEE IF THERE IS ALREADY AN
060376  001402                          BEQ     RK5H1               ;RK05 FUNCTION IN PROGRESS
060400  104000                          ERROR
060402  000000                          HALT

060404  012737  000340  177776  RK5H1:  MOV     #340,@#PSW          ;RAISE THE PRIORITY
060412  011637  060342                  MOV     (SP),RK5TMP
060416  062716  000016                  ADD     #16,(SP)
060422  104412                          SAVREG
060424  013700  060342                  MOV     RK5TMP,R0
060430  112037  060344                  MOVB    (R0)+,RK5FUN        ;GET THE ARGUMENTS.
060434  112037  060346                  MOVB    (R0)+,RK5UNI
060440  012037  060350                  MOV     (R0)+,RK5DA1
```

```
060444  012037  060352                    MOV     (R0)+,RK5DA2
060450  012037  060354                    MOV     (R0)+,RK5MA1
060454  012037  060356                    MOV     (R0)+,RK5MA2
060460  012037  060360                    MOV     (R0)+,RK5WCT
060464  012037  060362                    MOV     (R0)+,RK5VEC

060470  005037  060330                    CLR     RK5ER1           ;CLR THE ERROR FLAGS
060474  005037  060332                    CLR     RK5ER2
060500  005037  060334                    CLR     RK5ER3

060504  004737  060754                    JSR     PC,RK5S1         ;SET UP THE DRIVE NUMBER
060510  004737  061160                    JSR     PC,RK5RDY        ;GET THE DEVICE AND CONTROL
                                                                   ;READY
060514  004737  060776                    JSR     PC,RK5S2         ;COMPUTE THE SURFACE
                                                                   ;CYLINDER AND SECTOR
                                                                   ;ADDRESS.
060520  004737  061100                    JSR     PC,RK5S3         ;SET UP A WORD COUNT,
                                                                   ;THE UNIBUS MAP
                                                                   ;AND BUS ADDRESS.

060524  005077  123320            RK5H2:  CLR     @RK5CS1
060530  013777  060346  123320            MOV     RK5UNI,@RK5DA    ;SET THE DEVICE REGISTERS
060536  013777  060360  123306            MOV     RK5WCT,@RK5WC    ;TO DO THE FUNCTION
060544  013777  060354  123302            MOV     RK5MA1,@RK5BA
060552  053777  060356  123270            BIS     RK5MA2,@RK5CS1
060560  053777  060350  123270            BIS     RK5DA1,@RK5DA
060566  013700  004106                    MOV     RK5V,R0          ;LOAD THE INTERRUPT VECTOR
060572  012720  060644                    MOV     #RK5H4,(R0)+
060576  012710  000340                    MOV     #340,(R0)
060602  013700  060344                    MOV     RK5FUN,R0
060606  010037  060326                    MOV     R0,RK5FLG
060612  050077  123232                    BIS     R0,@RK5CS1       ;LOAD THE FUNCTION AND
                                                                   ;GO

060616  032700  000100                    BIT     #BIT6,R0         ;SEE IF THE FUNCTION WILL
060622  001402                            BEQ     RK5H3            ;INTERRUPT WHEN DONE.
060624  104414                            RESREG                   ;IF YES RETURN
060626  000002                            RTI

060630  004737  060672            RK5H3:  JSR     PC,RK5H5         ;IF THE FUNCTION WAS
060634  005037  060326                    CLR     RK5FLG           ;NOT INTERRUPT ENABLED
060640  104414                            RESREG                   ;WAIT FOR DONE OR ERROR.
060642  000002                            RTI

060644  004737  060672            RK5H4:  JSR     PC,RK5H5         ;SEE IF THERE WERE ANY ERRORS.
060650  005037  060326                    CLR     RK5FLG
060654  012777  060670  123224            MOV     #1$,@RK5V
060662  000230                            SPL     0
060664  000177  177472                    JMP     @RK5VEC
060670  000002                    1$:     RTI

060672  010046                    RK5H5:  MOV     R0,-(SP)
060674  017700  123150            RK5H51: MOV     @RK5CS1,R0       ;SEE IF ANY ERROR OCCURRED
060700  005700                            TST     R0
060702  100015                            BPL     RK5H6
060704  017737  123136  060332            MOV     @RK5ER,RK5ER2    ;IF YES, FLAG THE ERROR
```

```
060712  017737  123126  060334           MOV     @RK5DS,RK5ER3    ;AND SAVE THE STATUS
060720  012737  177777  060330           MOV     #-1,RK5ER1
060726  004737  061320                   JSR     PC,RK5CLR
060732  012600                           MOV     (SP)+,R0
060734  000207                           RTS     PC

060736  105700                   RK5H6:  TSTB    R0               ;WAIT FOR DONE OR
060740  100355                           BPL     RK5H51           ;ERROR
060742  105777  123076                   TSTB    @RK5DS
060746  100352                           BPL     RK5H51
060750  012600                           MOV     (SP)+,R0
060752  000207                           RTS     PC

060754  013700  060346           RK5S1:  MOV     RK5UNI,R0
060760  072027  000015                   ASH     #13.,R0
060764  042700  017777                   BIC     #017777,R0
060770  010037  060346                   MOV     R0,RK5UNI
060774  000207                           RTS     PC

060776  013701  060350           RK5S2:  MOV     RK5DA1,R1        ;COMPUTE THE CYLINDER
061002  005000                           CLR     R0               ;SURFACE AND SECTOR
061004  071027  011100                   DIV     #4672.,R0        ;DISK ADDRESS
061010  005000                           CLR     R0
061012  071027  000030                   DIV     #24.,R0
061016  010002                           MOV     R0,R2
061020  005000                           CLR     R0
061022  071027  000014                   DIV     #12.,R0
061026  010237  060370                   MOV     R2,RK5CYL
061032  010137  060366                   MOV     R1,RK5SEC
061036  010037  060364                   MOV     R0,RK5TRK
061042  072227  000005                   ASH     #5,R2
061046  042702  160037                   BIC     #160037,R2
061052  072027  000004                   ASH     #4,R0
061056  042700  177757                   BIC     #177757,R0
061062  042701  177760                   BIC     #177760,R1
061066  050100                           BIS     R1,R0
061070  050200                           BIS     R2,R0
061072  010037  060350                   MOV     R0,RK5DA1
061076  000207                           RTS     PC

061100  005437  060360           RK5S3:  NEG     RK5WCT           ;COMPUTE A VALID
                                                                  ;WORD COUNT AND
061104  013700  060354                   MOV     RK5MA1,R0        ;SET THE UB MAP
061110  013701  060356                   MOV     RK5MA2,R1        ;REGISTERS
061114  042701  177700                   BIC     #177700,R1
061120  012702  170300                   MOV     #MAPL20,R2
061124  012703  000010                   MOV     #10,R3
061130  010022                   1$:     MOV     R0,(R2)+
061132  010122                           MOV     R1,(R2)+
061134  062700  020000                   ADD     #20000,R0
061140  005501                           ADC     R1
061142  077306                           SOB     R3,1$
061144  012737  000040  060356           MOV     #40,RK5MA2
061152  005037  060354                   CLR     RK5MA1
061156  000207                           RTS     PC

```

```
061160  053777  060346  122670  RK5RDY: BIS     RK5UNI,@RK5DA     ;DO A CONTROL CLEAR
061166  012777  000001  122654          MOV     #1,@RK5CS1        ;FUNCTION
061174  105777  122650          1$:     TSTB    @RK5CS1
061200  100375                          BPL     1$

061202  053777  060346  122646          BIS     RK5UNI,@RK5DA     ;DO A DRIVE CLEAR
061210  012777  000015  122632          MOV     #15,@RK5CS1       ;FUNCTION

061216  017701  122626          2$:     MOV     @RK5CS1,R1        ;WAIT FOR DONE OR
061222  100420                          BMI     5$                ;ERROR.
061224  105701                          TSTB    R1
061226  100373                          BPL     2$

061230  017701  122610          3$:     MOV     @RK5DS,R1
061234  032701  000040                  BIT     #BIT5,R1          ;WRITE ENABLED?
061240  001011                          BNE     5$
061242  005777  122600                  TST     @RK5ER
061246  100406                          BMI     5$
061250  105701                          TSTB    R1
061252  100366                          BPL     3$
061254  032701  000100                  BIT     #BIT6,R1
061260  001763                          BEQ     3$
061262  000207                  4$:     RTS     PC

061264  005726                  5$:     TST     (SP)+
061266  017737  122554  060332          MOV     @RK5ER,RK5ER2
061274  017737  122544  060334          MOV     @RK5DS,RK5ER3
061302  012737  177777  060330          MOV     #-1,RK5ER1
061310  004737  061320                  JSR     PC,RK5CLR
061314  104414                          RESREG
061316  000002                          RTI

061320  005077  122532          RK5CLR: CLR     @RK5DA            ;RESET THE CONTROLLER
061324  012777  000001  122516          MOV     #1,@RK5CS1        ;BY DOING A CONTROL
061332  105777  122512          1$:     TSTB    @RK5CS1           ;CLEAR FUNCTION
061336  100375                          BPL     1$
061340  000207                          RTS     PC

                                        .SBTTL          UNIBUS EXERCISER HANDLER
                                ;UNIBUS EXERCISER HANDLER

                                ;REGISTERS USED IN UBEHAN
061342  000000                  UBEFLG:.WORD    0
061344  000000                  UBEER1:.WORD    0                 ;ERROR FLAGS.
061346  000000                  UBEER2:.WORD    0
061350  000000                  UBEER3:.WORD    0
061352  000000                  UBEER4:.WORD    0
061354  000000                  UBEUSE:.WORD    0
061356  000000                  UBETMP:.WORD    0
061360  000000                  UBEFUN:.WORD    0
061362  000000                  UBEUNI:.WORD    0
061364  000000                  UBEDA1:.WORD    0
061366  000000                  UBEDA2:.WORD    0
061370  000000                  UBEMA1:.WORD    0
061372  000000                  UBEMA2:.WORD    0
061374  000000                  UBEWCT:.WORD    0
```

```
061376  000000                  UBEVEC:.WORD    0
061400  000000                  UBETRK:.WORD    0
061402  000000                  UBESEC:.WORD    0
061404  000000                  UBECYL:.WORD    0

061406  005737  061342          UBEHAN: TST     UBEFLG            ;SEE IF THERE IS ALREADY
061412  001402                          BEQ     UBEH1             ;A UNIBUS EXERCISER FUNCTION
061414  104000                          ERROR                     ;IN PROGRESS. IF THERE
061416  000000                          HALT                      ;IS ERROR. (SHOULD NEVER HAPPEN)

061420  012737  000340  177776  UBEH1:  MOV     #340,@#PSW        ;RAISE THE PRIORITY
061426  011637  061356                  MOV     (SP),UBETMP       ;GET AN ARGUMENT POINTER
061432  062716  000016                  ADD     #16,(SP)
061436  104412                          SAVREG
061440  013700  061356                  MOV     UBETMP,R0         ;RESET THE RETURN ADDRESS

061444  012037  061360                  MOV     (R0)+,UBEFUN      ;GET THE ARGUMENTS.
061450  012037  061364                  MOV     (R0)+,UBEDA1
061454  012037  061366                  MOV     (R0)+,UBEDA2
061460  012037  061370                  MOV     (R0)+,UBEMA1
061464  012037  061372                  MOV     (R0)+,UBEMA2
061470  012037  061374                  MOV     (R0)+,UBEWCT
061474  012037  061376                  MOV     (R0)+,UBEVEC
061500  005037  061344                  CLR     UBEER1            ;CLEAR THE ERROR FLAGS
061504  005037  061346                  CLR     UBEER2
061510  005037  061350                  CLR     UBEER3
061514  004737  062004                  JSR     PC,UBERDY
061520  004737  061722                  JSR     PC,UBES1                   ;GO SET UP THE BUS
                                                                  ;ADDRESS AND UB MAP

061524  013777  061374  122334  UBEH2:  MOV     UBEWCT,@UBECC     ;SET THE DEVICE
061532  013777  061370  122330          MOV     UBEMA1,@UBEBA     ;REGISTERS
061540  053777  061372  122330          BIS     UBEMA2,@UBECR2
061546  013777  061366  122310          MOV     UBEDA2,@UBEDB
061554  013700  004110                  MOV     UBEV,R0
061560  012720  061632                  MOV     #UBEH4,(R0)+
061564  012710  000340                  MOV     #340,(R0)
061570  013700  061360                  MOV     UBEFUN,R0
061574  010037  061342                  MOV     R0,UBEFLG
061600  010077  122266                  MOV     R0,@UBECR1        ;LOAD THE FUNCTION
061604  032700  000100                  BIT     #BIT6,R0          ;SEE IF THE FUNCTION
061610  001402                          BEQ     UBEH3             ;IS INTERRUPT ENABLED
061612  104414                          RESREG                    ;IF YES RETURN
061614  000002                          RTI

061616  004737  061646          UBEH3:  JSR     PC,UBEH5          ;IF NOT INTERRUPT ENABLED
061622  005037  061342                  CLR     UBEFLG            ;WAIT FOR DONE OR
061626  104414                          RESREG                    ;ERROR
061630  000002                          RTI

061632  005037  061342          UBEH4:  CLR     UBEFLG            ;WHEN THE INTERRUPT
061636  004737  061646                  JSR     PC,UBEH5          ;OCCURS SEE IF ANY ERRORS
061642  000177  177530                  JMP     @UBEVEC           ;OCCURRED

061646  010046                  UBEH5:  MOV     R0,-(SP)
061650  017700  122216          UBEH51: MOV     @UBECR1,R0        ;WAIT FOR DONE OR
```

```
061654  005700                            TST     R0              ;ERROR
061656  100015                            BPL     UBEH6

061660  017737  122206  061346            MOV     @UBECR1,UBEER2
061666  017737  122204  061350            MOV     @UBECR2,UBEER3
061674  012737  177777  061344            MOV     #-1,UBEER1
061702  004737  062060                    JSR     PC,UBCLR
061706  012600                            MOV     (SP)+,R0
061710  000207                            RTS     PC

061712  105700                  UBEH6:    TSTB    R0
061714  100355                            BPL     UBEH51
061716  012600                            MOV     (SP)+,R0
061720  000207                            RTS     PC

061722  013700  061370          UBES1:    MOV     UBEMA1,R0       ;SET UP THE BUS ADDRESS
061726  013701  061372                    MOV     UBEMA2,R1       ;AND UB MAPPING BOX
061732  042701  177700                    BIC     #177700,R1
061736  012702  170200                    MOV     #MAPL00,R2
061742  012703  000010                    MOV     #10,R3

061746  010022                  1$:       MOV     R0,(R2)+
061750  010122                            MOV     R1,(R2)+
061752  062700  020000                    ADD     #20000,R0
061756  005501                            ADC     R1
061760  077306                            SOB     R3,1$

061762  005037  061372                    CLR     UBEMA2
061766  005037  061370                    CLR     UBEMA1
061772  005137  061374                    COM     UBEWCT
061776  005237  061374                    INC     UBEWCT
062002  000207                            RTS     PC

062004  005077  122064          UBERDY:   CLR     @UBECLR         ;TRY TO GET DEVICE
                                                                  ;READY
062010  017700  122056          1$:       MOV     @UBECR1,R0
062014  100403                            BMI     2$
062016  105700                            TSTB    R0
062020  100373                            BPL     1$
062022  000207                            RTS     PC

062024  005726                  2$:       TST     (SP)+
062026  017737  122040  061346            MOV     @UBECR1,UBEER2
062034  017737  122036  061350            MOV     @UBECR2,UBEER3
062042  012737  177777  061350            MOV     #-1,UBEER3
062050  004737  062060                    JSR     PC,UBCLR
062054  104414                            RESREG
062056  000002                            RTI

062060  005077  122010          UBCLR:    CLR     @UBECLR         ;CLEAR THE DEVICE.
062064  105777  122002          1$:       TSTB    @UBECR1
062070  100375                            BPL     1$
062072  000207                            RTS     PC

                                          .SBTTL          MASS BUS TESTER HANDLER
                                ;THIS CODE IS FOR HANDLING THE MASS BUS
```

```
                                                ;TESTED DEVICE.

                                                ;REGISTERS USED IN RH4HAN
062074  000000                                  RH4FLG:.WORD    0
062076  000000                                  RH4ER1:.WORD    0                       ;ERROR FLAGS.
062100  000000                                  RH4ER2:.WORD    0
062102  000000                                  RH4ER3:.WORD    0
062104  000000                                  RH4ER4:.WORD    0
062106  000000                                  RH4USE:.WORD    0
062110  000000                                  RH4TMP:.WORD    0
062112  000000                                  RH4FUN:.WORD    0
062114  000000                                  RH4UNI:.WORD    0
062116  000000                                  RH4DA1:.WORD    0
062120  000000                                  RH4DA2:.WORD    0
062122  000000                                  RH4MA1:.WORD    0
062124  000000                                  RH4MA2:.WORD    0
062126  000000                                  RH4WCT:.WORD    0
062130  000000                                  RH4VEC:.WORD    0
062132  000000                                  RH4TRK:.WORD    0
062134  000000                                  RH4SEC:.WORD    0
062136  000000                                  RH4CYL:.WORD    0

062140  005737  062074                          RH4HAN: TST     RH4FLG                  ;SEE IF A FUNCTION
062144  001402                                          BEQ     RH4H1                   ;IS ALREADY ACTIVE IF
062146  104000                                          ERROR                           ;SO ERROR.
062150  000000                                          HALT

062152  012777  000340  115616                  RH4H1:  MOV     #340,@PSW               ;RAISE THE PRIORITY
062160  011637  062110                                  MOV     (SP),RH4TMP
062164  062716  000016                                  ADD     #16,(SP)
062170  104412                                          SAVREG
062172  013700  062110                                  MOV     RH4TMP,R0               ;RESET THE RETURN
062176  112037  062112                                  MOVB    (R0)+,RH4FUN
062202  112037  062114                                  MOVB    (R0)+,RH4UNI
062206  012037  062116                                  MOV     (R0)+,RH4DA1
062212  012037  062120                                  MOV     (R0)+,RH4DA2
062216  012037  062122                                  MOV     (R0)+,RH4MA1
062222  012037  062124                                  MOV     (R0)+,RH4MA2
062226  012037  062126                                  MOV     (R0)+,RH4WCT
062232  011037  062130                                  MOV     (R0),RH4VEC
062236  005037  062076                                  CLR     RH4ER1                  ;CLEAR THE ERROR FLAGS
062242  005037  062100                                  CLR     RH4ER2
062246  005037  062102                                  CLR     RH4ER3
062252  004737  062532                                  JSR     PC,RH4S1                ;SET UP THE UNIT NUMBER
062256  004737  062556                                  JSR     PC,RH4RDY               ;GET THE UNIT READY
062262  004737  062542                                  JSR     PC,RH4S2

062266  013777  062114  121520                  RH4H2:  MOV     RH4UNI,@RH4CS2          ;SET THE CONTROL REGISTERS
062274  013777  062126  121504                          MOV     RH4WCT,@RH4WC           ;AND DEVICE REGISTERS
062302  013777  062122  121500                          MOV     RH4MA1,@RH4BA
062310  013777  062124  121520                          MOV     RH4MA2,@RH4AE
062316  013777  062116  121500                          MOV     RH4DA1,@RH4DR
062324  012777  004000  121476                          MOV     #4000,@RH4MR1
062332  000240                                          NOP
062334  013700  004104                                  MOV     RH4V,R0                 ;VECTOR
062340  012720  062412                                  MOV     #RH4H4,(R0)+
062344  012710  000340                      MOV     #340,(R0)
062350  013700  062112                      MOV     RH4FUN,R0
062354  010037  062074                      MOV     R0,RH4FLG           ;LOAD THE FUNCTION AND
062360  110077  121420                      MOVB    R0,@RH4CS1          ;GO
062364  032700  000100                      BIT     #BIT6,R0            ;SEE IF THIS FUNCTION
062370  001402                              BEQ     RH4H3               ;WILL INTERRUPT WHEN DONE
062372  104414                              RESREG                      ;IF YES RETURN TO CALL
062374  000002                              RTI

062376  004737  062426            RH4H3:    JSR     PC,RH4H5            ;IF NOT INTERRUPT
062402  005037  062074                      CLR     RH4FLG              ;ENABLED WAIT FOR
062406  104414                              RESREG                      ;THE FUNCTION TO
062410  000002                              RTI                         ;FINISH THEN RETURN.

062412  005037  062074            RH4H4:    CLR     RH4FLG              ;WHEN THE INTERRUPT
062416  004737  062426                      JSR     PC,RH4H5            ;OCCURS CHECKS FOR
062422  000177  177502                      JMP     @RH4VEC             ;ERRORS. THEN GO TO THE
                                                                        ;SPECIFIED SERVICE
                                                                        ;ROUTINE

062426  010046                    RH4H5:    MOV     R0,-(SP)
062430  053777  062114  121356    RH4H51:   BIS     RH4UNI,@RH4CS2
062436  017700  121342                      MOV     @RH4CS1,R0          ;SEE IF THE FUNCTION
062442  005700                              TST     R0                  ;WAS COMPLETED WITHOUT
062444  100023                              BPL     RH4H6               ;ERRORS.
062446  032700  060000                      BIT     #60000,R0
062452  001420                              BEQ     RH4H6
062454  017737  121334  062100              MOV     @RH4CS2,RH4ER2      ;IF ERRORS OCCURRED
062462  017737  121330  062102              MOV     @RH4ST,RH4ER3       ;SAVE STATUS AND SET
062470  017737  121324  062104              MOV     @RH4ER,RH4ER4
062476  012737  177777  062076              MOV     #-1,RH4ER1          ;ERROR FLAGS.
062504  004737  062750                      JSR     PC,RH4CLR
062510  012600                              MOV     (SP)+,R0
062512  000207                              RTS     PC

062514  105700                    RH4H6:    TSTB    R0                  ;WAIT FOR READY OR
062516  100344                              BPL     RH4H51              ;ERROR
062520  105777  121272                      TSTB    @RH4ST
062524  100341                              BPL     RH4H51
062526  012600                              MOV     (SP)+,R0
062530  000207                              RTS     PC

062532  042737  177770  062114    RH4S1:    BIC     #177770,RH4UNI      ;SET UP THE DRIVE NUMBER
062540  000207                              RTS     PC

062542  012737  000000  062120    RH4S2:    MOV     #0,RH4DA2           ;FOR DEBUG.
062550  005437  062126                      NEG     RH4WCT              ;SET UP WORD COUNT
062554  000207                              RTS     PC

062556  012737  000040  062106    RH4RDY:   MOV     #BIT5,RH4USE        ;CLR THE CONTROLLER
062564  053737  062114  062106              BIS     RH4UNI,RH4USE
062572  013777  062106  121214              MOV     RH4USE,@RH4CS2
062600  013777  062114  121206              MOV     RH4UNI,@RH4CS2
062606  105777  121172            1$:       TSTB    @RH4CS1 ;AND DRIVES
062612  100375                              BPL     1$
062614  013777  062114  121172              MOV     RH4UNI,@RH4CS2      ;DO A NOP FUNCTION
```

```
062622  012777  000001  121154           MOV     #1,@RH4CS1        ;TO INITIALIZE THE
                                                                   ;DRIVE
062630  017701  121150          2$:      MOV     @RH4CS1,R1        ;WAIT FOR READY OR ERROR.
062634  005701                           TST     R1
062636  100420                           BMI     4$
062640  105701                           TSTB    R1
062642  100372                           BPL     2$

062644  017700  121146          3$:      MOV     @RH4ST,R0         ;LOOK AT THE UNIT STATUS
062650  032700  000400                   BIT     #BIT8,R0          ;UNIT PRESENT?
062654  001414                           BEQ     5$
062656  032700  010000                   BIT     #BIT12,R0         ;ON LINE?
062662  001411                           BEQ     5$
062664  032700  040000                   BIT     #BIT14,R0         ;ANY ERRORS?
062670  001006                           BNE     5$
062672  105700                           TSTB    R0                ;WAIT FOR UNIT READY
062674  100355                           BPL     2$
062676  000207                           RTS     PC

062700  032701  040000          4$:      BIT     #BIT14,R1         ;ATTENTION OR ERROR
062704  001757                           BEQ     3$
062706  005726                  5$:      TST     (SP)+             ;FLAG AND RECORD ERROR
062710  017737  121100  062100           MOV     @RH4CS2,RH4ER2
062716  017737  121074  062102           MOV     @RH4ST,RH4ER3
062724  017737  121070  062104           MOV     @RH4ER,RH4ER4
062732  012737  177777  062076           MOV     #-1,RH4ER1
062740  004737  062750                   JSR     PC,RH4CLR
062744  104414                           RESREG
062746  000002                           RTI

062750  013777  062106  121036  RH4CLR:  MOV     RH4USE,@RH4CS2    ;CLR THE CONTROLLER
062756  105777  121022          1$:      TSTB    @RH4CS1           ;AND DRIVES.
062762  100375                           BPL     1$
062764  000207                  TSTDT1:  RTS     PC
062766  001000                           .BLKW   512.
                                ;SPECIAL MESSAGES:

064766  041536  000200          CONCMS:  .ASCIZ  '^C'<CRLF>

064772  047515  044516  047524  MMESRS:  .ASCIZ  'MONITOR (OR LOADER) RESTORED!'<CRLF>
065000  020122  047450  020122
065006  047514  042101  051105
065014  020051  042522  052123
065022  051117  042105  100041
065030     000

065031     200  047520  042527  POWERM:  .ASCIZ  <CRLF>'POWER FAILURE, PROGRAM RESTARTING'<CRLF><CRLF>
065036  020122  040506  046111
065044  051125  026105  050040
065052  047522  051107  046501
065060  051040  051505  040524
065066  052122  047111  100107
065074  000200

065076  000011                  $TAB:    .ASCIZ  <TAB>

```

```
065100 042600 050130 041505 MTA5:   .ASCII  <CRLF>'EXPECTED DATA:'<CRLF>
065106 042524 020104 040504
065114 040524 100072
065120 051107 052517 020120         .ASCIZ  'GROUP 0.GROUP 1.MEM EV.'<TAB>'MEM ODD.'<CRLF>
065126 027060 051107 052517
065134 020120 027061 042515
065142 020115 053105 004456
065150 042515 020115 042117
065156 027104 000200

065162 042200 052101 020101 MTA11:  .ASCII  <CRLF>'DATA WRITTEN.'<TAB>'TEST ADDR.'<TAB>'ERROR REG.'<CRLF>
065170 051127 052111 042524
065176 027116 052011 051505
065204 020124 042101 051104
065212 004456 051105 047522
065220 020122 042522 027107
065226    200

065227    040 047111 000040 MTA17:  .ASCIZ  ' IN '

065234 054105 042520 052103 MTB17:  .ASCIZ  'EXPECTED DATA:'<CRLF>
065242 042105 042040 052101
065250 035101 000200

065254 054502 042524 004456 MTC17:  .ASCIZ  'BYTE.'<TAB>
065262    000

065263    127 051117 027104 MTA20:  .ASCIZ  'WORD.'<TAB>
065270 000011

065272 054105 042520 052103 MTA21:  .ASCII  'EXPECTED DATA:'<CRLF>
065300 042105 042040 052101
065306 035101    200
065311    110 052111 020123         .ASCIZ  'HITS IN GROUP 0.'<TAB>'/'<TAB>'HITS IN GROUP 1. '<CRLF>
065316 047111 043440 047522
065324 050125 030040 004456
065332 004457 044510 051524
065340 044440 020116 051107
065346 052517 020120 027061
065354 100040    000

       065227               MTB21=MTA17

065357    200 042524 052123 MTA43:  .ASCII  <CRLF>'TEST ADDRESS.'<TAB>'ERROR ADRS REG.'<TAB>
065364 040440 042104 042522
065372 051523 004456 051105
065400 047522 020122 042101
065406 051522 051040 043505
065414 004456
065416 051105 047522 020122         .ASCIZ  'ERROR REG.'<CRLF>
065424 042522 027107 000200

065432 053600 047522 042524 MTA45:  .ASCIZ  <CRLF>'WROTE. 377'<TAB>'IN BYTE. '
065440 020056 033463 004467
065446 047111 041040 052131
065454 027105 000040
```

```

065460  051200  040505  020104  MTB45:  .ASCIZ  <CRLF>'READ DATA. '
065466  040504  040524  020056
065474     000

065475     011  047111  053440  MTC45:  .ASCIZ  <TAB>'IN WORD. '
065502  051117  027104  000040

065510  053600  047522  042524  MTA50:  .ASCIZ  <CRLF>'WROTE. 000'<TAB>'IN BYTE. '
065516  020056  030060  004460
065524  047111  041040  052131
065532  027105  000040

065536  042600  052116  051105  PDMSG1: .ASCII  <CRLF>'ENTERING CACHE ADDRESS MEMORY POWER UP '
065544  047111  020107  040503
065552  044103  020105  042101
065560  051104  051505  020123
065566  042515  047515  054522
065574  050040  053517  051105
065602  052440  020120
065606  047111  040526  044514          .ASCII  'INVALIDATOR TEST.'<CRLF>
065614  040504  047524  020122
065622  042524  052123  100056
065630  046120  040505  042523          .ASCII  'PLEASE GO THROUGH A POWER DOWN, POWER UP '
065636  043440  020117  044124
065644  047522  043525  020110
065652  020101  047520  042527
065660  020122  047504  047127
065666  020054  047520  042527
065674  020122  050125     040
065701     123  050505  042525          .ASCIZ  'SEQUENCE.'<CRLF>
065706  041516  027105  000200

065714  041600  041501  042510  PDMSG2: .ASCII  <CRLF>'CACHE ADDRESS MEMORY POWER UP INVALIDATOR'
065722  040440  042104  042522
065730  051523  046440  046505
065736  051117  020131  047520
065744  042527  020122  050125
065752  044440  053116  046101
065760  042111  052101  051117
065766  052040  051505  020124          .ASCIZ  ' TEST DID NOT FAIL.'<CRLF>
065774  044504  020104  047516
066002  020124  040506  046111
066010  100056     000

066013     105  051122  051117  ADRNG:  .ASCII  'ERROR ADDRESS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
066020  040440  042104  042522
066026  051523  051040  043505
066034  051511  042524  020122
066042  042516  042105  042105
066050  043040  051117  052040
066056  051505  026124  041200
066064  052125  044440  020124
066072  040510  020123  042502
066100  047105     040
066103     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
```

```
066110  042105  040440  020123
066116  040502  020504     000

066123     105  051122  051117  ERRNG:  .ASCII  'ERROR REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
066130  051040  043505  051511
066136  042524  020122  042516
066144  042105  042105  043040
066152  051117  052040  051505
066160  026124  041200  052125
066166  044440  020124  040510
066174  020123  042502  047105
066202     040
066203     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
066210  042105  040440  020123
066216  040502  020504     000

066223     103  047117  051124  CNRNG:  .ASCII  'CONTROL REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
066230  046117  051040  043505
066236  051511  042524  020122
066244  042516  042105  042105
066252  043040  051117  052040
066260  051505  026124  041200
066266  052125  044440  020124
066274  040510  020123  042502
066302  047105     040
066305     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
066312  042105  040440  020123
066320  040502  020504     000
066325     115  044501  052116  MNRNG:  .ASCII  'MAINTENANCE REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
066332  047105  047101  042503
066340  051040  043505  051511
066346  042524  020122  042516
066354  042105  042105  043040
066362  051117  052040  051505
066370  026124  041200  052125
066376  044440  020124  040510
066404  020123  042502  047105
066412     040
066413     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
066420  042105  040440  020123
066426  040502  020504     000

066433     110  052111  046457  HMRNG:  .ASCII  'HIT/MISS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
066440  051511  020123  042522
066446  044507  052123  051105
066454  047040  042505  042504
066462  020104  047506  020122
066470  042524  052123  100054
066476  052502  020124  052111
066504  044040  051501  041040
066512  042505  020116
066516  046106  043501  042507          .ASCIZ  'FLAGGED AS BAD!'
066524  020104  051501  041040
066532  042101  000041

066536  040600  042104  042522  MTA77:  .ASCIZ  <CRLF>'ADDRESS:  '
```

```
066544  051523  020072  000040

066552  051440  047510  046125  MTB77:  .ASCIZ  ' SHOULD HAVE BEEN A HIT IN GROUP '
066560  020104  040510  042526
066566  041040  042505  020116
066574  020101  044510  020124
066602  047111  043440  047522
066610  050125  000040

066614  043101  042524  020122  MTC77:  .ASCIZ  'AFTER REFERENCING'<CRLF>'ADDRESS:  '
066622  042522  042506  042522
066630  041516  047111  100107
066636  042101  051104  051505
066644  035123  020040     000

066651     040  044127  046111  MTD77:  .ASCIZ  ' WHILE FORCING SELECTION OF GROUP '
066656  020105  047506  041522
066664  047111  020107  042523
066672  042514  052103  047511
066700  020116  043117  043440
066706  047522  050125  000040

066714  040600  051122  051117  MTA101: .ASCII  <CRLF>'ARROR ADRS REG.'<TAB>'ERROR REG.'<TAB>
066722  040440  051104  020123
066730  042522  027107  042411
066736  051122  051117  051040
066744  043505  004456
066750  054105  042520  052103          .ASCIZ  'EXPECTED ERR.'<TAB>'PATTERN PUT IN MAINT REG.'<CRLF>
066756  042105  042440  051122
066764  004456  040520  052124
066772  051105  020116  052520
067000  020124  047111  046440
067006  044501  052116  051040
067014  043505  100056     000

067021     200  043101  042524  MTA120: .ASCIZ  <CRLF>'AFTER 2ND CYCLE READ  '
067026  020122  047062  020104
067034  054503  046103  020105
067042  042522  042101  020040
067050     000

067051     200  043101  042524  MTB120: .ASCIZ  <CRLF>'AFTER 4TH CYCLE READ  '
067056  020122  052064  020110
067064  054503  046103  020105
067072  042522  042101  020040
067100     000

067101     200  043101  042524  MTC120: .ASCIZ  <CRLF>'AFTER 6TH CYCLE READ  '
067106  020122  052066  020110
067114  054503  046103  020105
067122  042522  042101  020040
067130     000
067131     200  043101  042524  MTD120: .ASCIZ  <CRLF>'AFTER 8TH CYCLE READ  '
067136  020122  052070  020110
067144  054503  046103  020105
067152  042522  042101  020040
```

```
067160     000

067161     200  043101  042524  MTE120: .ASCIZ  <CRLF>'AFTER 10TH CYCLE READ '
067166  020122  030061  044124
067174  041440  041531  042514
067202  051040  040505  020104
067210     000

067211     200  043101  042524  MTF120: .ASCIZ  <CRLF>'AFTER 12TH CYCLE READ '
067216  020122  031061  044124
067224  041440  041531  042514
067232  051040  040505  020104
067240     000

067241     106  047522  020115  MTG120: .ASCIZ  'FROM THE HIT/MISS REG. EXPECTED '
067246  044124  020105  044510
067254  027524  044515  051523
067262  051040  043505  020056
067270  054105  042520  052103
067276  042105  000040

067302  052200  042510  050040  MTA124: .ASCII  <CRLF>'THE PATTERN BEING USED IN THE MAINTENANCE '
067310  052101  042524  047122
067316  041040  044505  043516
067324  052440  042523  020104
067332  047111  052040  042510
067340  046440  044501  052116
067346  047105  047101  042503
067354     040
067355     122  043505  051511          .ASCIZ  'REGISTER WAS: '
067362  042524  020122  040527
067370  035123  000040

067374  051200  043105  051105  MTA126: .ASCIZ  <CRLF>'REFERENCED ADDRESS:'<TAB>
067402  047105  042503  020104
067410  042101  051104  051505
067416  035123  000011

067422  040600  051122  051117  MTB126: .ASCIZ  <CRLF>'ARROR ADDRESS REGISTER:'<TAB>
067430  040440  042104  042522
067436  051523  051040  043505
067444  051511  042524  035122
067452  000011

067454  050200  052101  042524  MTA131: .ASCIZ  <CRLF>'PATTERN BEING USED IN THE MAINTENANCE REGISTER:'<TAB>
067462  047122  041040  044505
067470  043516  052440  042523
067476  020104  047111  052040
067504  042510  046440  044501
067512  052116  047105  047101
067520  042503  051040  043505
067526  051511  042524  035122
067534  000011

067536  042600  050130  041505  MTB131: .ASCIZ  <CRLF>'EXPECTED ERROR REGISTER:'<TAB>
067544  042524  020104  051105
```

```
067552  047522  020122  042522
067560  044507  052123  051105
067566  004472     000

067571     200  047507  020124  MTC131: .ASCIZ  <CRLF>'GOT ERROR REGISTER:'<TAB>
067576  051105  047522  020122
067604  042522  044507  052123
067612  051105  004472     000

067617     200  051105  047522  MTA134: .ASCIZ  <CRLF>'ERROR ADR REG.'<TAB>'ERROR REG.'<CRLF>
067624  020122  042101  020122
067632  042522  027107  042411
067640  051122  051117  051040
067646  043505  100056     000

067653     200  054105  042520  MTA135: .ASCIZ  <CRLF>'EXPECTED ERROR REG.:  '
067660  052103  042105  042440
067666  051122  051117  051040
067674  043505  035056  020040
067702     000

067703     107  052117  042440  MTB135: .ASCIZ  'GOT ERROR REG.:  '
067710  051122  051117  051040
067716  043505  035056  020040
067724     000

067725     200  054105  042520  MTC135: .ASCIZ  <CRLF>'EXPECTED ERROR ADR REG.:  '
067732  052103  042105  042440
067740  051122  051117  040440
067746  051104  051040  043505
067754  035056  020040     000

067761     107  052117  042440  MTD135: .ASCIZ  'GOT ERROR ADR REG.:  '
067766  051122  051117  040440
067774  051104  051040  043505
070002  035056  020040     000


070007     015  053412  051101  MS01:   .ASCII  <15><12>/WARNING- THE SIZE OF MEMORY IS DIFFERENT THEN THAT/<CRLF>
070014  044516  043516  020055
070022  044124  020105  044523
070030  042532  047440  020106
070036  042515  047515  054522
070044  044440  020123  044504
070052  043106  051105  047105
070060  020124  044124  047105
070066  052040  040510  100124
070074  044440  042116  041511          .ASCIZ  / INDICATED BY THE SYSTEM SIZE REGISTER./
070102  052101  042105  041040
070110  020131  044124  020105
070116  054523  052123  046505
070124  051440  055111  020105
070132  042522  044507  052123
070140  051105  000056
070144  005015  044523  042532  MS02:   .ASCIZ  <15><12>/SIZE REG.     ACTUAL/
070152  051040  043505  020056
```

```
070160  020040  020040  041501
070166  052524  046101     000
070173     040  020040  020040  MS03:   .ASCIZ  /       /
070200  020040     000
070203     200  050103  020125  MSG1:   .ASCIZ<CRLF>    ''CPU UNDER TEST FOUND TO BE A ''
070210  047125  042504  020122
070216  042524  052123  043040
070224  052517  042116  052040
070232  020117  042502  040440
070240  000040
070242  041113  030461  042455  MSG2:   .ASCIZ  ''KB11-EM''<CRLF>
070250  100115     000
070253     113  030502  026461  MSG3:   .ASCIZ  ''KB11-B/C''<CRLF>
070260  027502  100103     000
070265     113  030502  026461  MSG4:   .ASCIZ  ''KB11-CM                 ''<CRLF>
070272  046503  020040  020040
070300  020040  020040  020040
070306  020040  020040  020040
070314  000200
070316  041113  030461  042455  MSG5:   .ASCIZ  ''KB11-E''<CRLF>
070324  000200
070326  005015  047516  046440  EM724:  .ASCIZ  <CR><LF>/NO MAP REGISTERS AVAILABLE FOR UNIBUS PARITY ERROR TEST/
070334  050101  051040  043505
070342  051511  042524  051522
070350  040440  040526  046111
070356  041101  042514  043040
070364  051117  052440  044516
070372  052502  020123  040520
070400  044522  054524  042440
070406  051122  051117  052040
070414  051505  000124


                                ;THESE ARE THE ERROR MESSAGES:

070420  020101  042522  042506  EM1:    .ASCIZ  'A REFERENCE WHICH SHOULD HAVE BEEN A HIT WAS A MISS.'
070426  042522  041516  020105
070434  044127  041511  020110
070442  044123  052517  042114
070450  044040  053101  020105
070456  042502  047105  040440
070464  044040  052111  053440
070472  051501  040440  046440
070500  051511  027123     000

070505     125  042516  050130  EM2:    .ASCII  'UNEXPECTED ERROR DURING WORST CASE NOISE TEST ON '
070512  041505  042524  020104
070520  051105  047522  020122
070526  052504  044522  043516
070534  053440  051117  052123
070542  041440  051501  020105
070550  047516  051511  020105
070556  042524  052123  047440
070564  020116
070566  040503  044103  020105          .ASCII  'CACHE DATA MEMORY.'<CRLF>
070574  040504  040524  046440
070602  046505  051117  027131
```

```
070610     200
070611     101  047040  047117             .ASCIZ  'A NON-CACHE DATA PARITY ERROR OCCURRED WHILE TESTING.'
070616  041455  041501  042510
070624  042040  052101  020101
070632  040520  044522  054524
070640  042440  051122  051117
070646  047440  041503  051125
070654  042522  020104  044127
070662  046111  020105  042524
070670  052123  047111  027107
070676     000

070677     127  051117  052123  EM3:   .ASCII  'WORST CASE NOISE TEST OF THE CACHE DATA MEMORY '
070704  041440  051501  020105
070712  047516  051511  020105
070720  042524  052123  047440
070726  020106  044124  020105
070734  040503  044103  020105
070742  040504  040524  046440
070750  046505  051117  020131
070756  043200  044501  042514         .ASCIZ  <CRLF>/FAILED WHILE GALLOPING 0'S./
070764  020104  044127  046111
070772  020105  040507  046114
071000  050117  047111  020107
071006  023460  027123     000

071013     127  051117  052123  EM4:   .ASCII  'WORST CASE NOISE TEST OF THE CACHE DATA MEMORY'
071020  041440  051501  020105
071026  047516  051511  020105
071034  042524  052123  047440
071042  020106  044124  020105
071050  040503  044103  020105
071056  040504  040524  046440
071064  046505  051117     131
071071     200  040506  046111         .ASCIZ  <CRLF>/FAILED WHILE GALLOPING 1'S./
071076  042105  053440  044510
071104  042514  043440  046101
071112  047514  044520  043516
071120  030440  051447  000056

071126  042103  054115  052040  EM5:   .ASCIZ  'CDMX TEST FAILURE.'<CRLF>'BAD CACHE GROUP 0 DATA READ.'
071134  051505  020124  040506
071142  046111  051125  027105
071150  041200  042101  041440
071156  041501  042510  043440
071164  047522  050125  030040
071172  042040  052101  020101
071200  042522  042101  000056

071206  042103  054115  052040  EM6:   .ASCIZ  'CDMX TEST FAILURE.'<CRLF>'BAD CACHE GROUP 1 DATA READ.'
071214  051505  020124  040506
071222  046111  051125  027105
071230  041200  042101  041440
071236  041501  042510  043440
071244  047522  050125  030440
071252  042040  052101  020101
```

```
071260  042522  042101  000056

071266  042103  054115  052040  EM7:    .ASCII  'CDMX TEST FAILURE.'<CRLF>'BAD MAIN MEMORY, EVEN WORD,'
071274  051505  020124  040506
071302  046111  051125  027105
071310  041200  042101  046440
071316  044501  020116  042515
071324  047515  054522  020054
071332  053105  047105  053440
071340  051117  026104
071344  042040  052101  020101          .ASCIZ  ' DATA READ.'
071352  042522  042101  000056

071360  042103  054115  052040  EM10:   .ASCII  'CDMX TEST FAILURE.'<CRLF>'BAD MAIN MEMORY, ODD WORD,'
071366  051505  020124  040506
071374  046111  051125  027105
071402  041200  042101  046440
071410  044501  020116  042515
071416  047515  054522  020054
071424  042117  020104  047527
071432  042122     054
071435     040  040504  040524          .ASCIZ  ' DATA READ.'
071442  051040  040505  027104
071450     000

071451     120  051101  052111  EM11:   .ASCIZ  'PARITY ERROR IN CACHE DATA MEMORY COUNT PATTERN TEST.'
071456  020131  051105  047522
071464  020122  047111  041440
071472  041501  042510  042040
071500  052101  020101  042515
071506  047515  054522  041440
071514  052517  052116  050040
071522  052101  042524  047122
071530  052040  051505  027124
071536     000

071537     102  042101  042040  EM12:   .ASCII  'BAD DATA WAS READ IN CACHE MEMORY COUNT PATTERN '
071544  052101  020101  040527
071552  020123  042522  042101
071560  044440  020116  040503
071566  044103  020105  042515
071574  047515  054522  041440
071602  052517  052116  050040
071610  052101  042524  047122
071616     040
071617     124  051505  027124          .ASCIZ  'TEST.'<CRLF>'BUT NO TRAP OR ABORT OCCURRED.'
071624  041200  052125  047040
071632  020117  051124  050101
071640  047440  020122  041101
071646  051117  020124  041517
071654  052503  051122  042105
071662  000056

071664  040503  044103  020105  EM13:   .ASCII  'CACHE MEMORY COUNT PATTERN TEST.'<CRLF>
071672  042515  047515  054522
071700  041440  052517  052116
```

```
11876  071706  050040  052101  042524
11877  071714  047122  052040  051505
11878  071722  027124     200
11879  071725     105  051122  051117            .ASCIZ  'ERROR SUMMARY.'
11880  071732  051440  046525  040515
11881  071740  054522  000056
11882
11883  071744  052600  042516  050130  EM14:     .ASCIZ  <CRLF>'UNEXPECTED PARITY ERROR TRAP.'
11884  071752  041505  042524  020104
11885  071760  040520  044522  054524
11886  071766  042440  051122  051117
11887  071774  052040  040522  027120
11888  072002     000
11889
11890  072003     052  025052  042524  EM15:     .ASCIZ  '***TEST ABORTED! GOING TO NEXT TEST.***'
11891  072010  052123  040440  047502
11892  072016  052122  042105  020041
11893  072024  047507  047111  020107
11894  072032  047524  047040  054105
11895  072040  020124  042524  052123
11896  072046  025056  025052     000
11897
11898  072053     103  041501  042510  EM16:     .ASCIZ  'CACHE DATA MEMORY DUAL ADDRESS TEST FAILED.'
11899  072060  042040  052101  020101
11900  072066  042515  047515  054522
11901  072074  042040  040525  020114
11902  072102  042101  051104  051505
11903  072110  020123  042524  052123
11904  072116  043040  044501  042514
11905  072124  027104     000
11906
11907  072127     103  041501  042510  EM17:     .ASCIZ  'CACHE DATA MEMORY BYTE ENABLE LOGIC TEST FAILED.'
11908  072134  042040  052101  020101
11909  072142  042515  047515  054522
11910  072150  041040  052131  020105
11911  072156  047105  041101  042514
11912  072164  046040  043517  041511
11913  072172  052040  051505  020124
11914  072200  040506  046111  042105
11915  072206  000056
11916
11917          072127                  EM20=EM17
11918
11919  072210  040503  044103  020105  EM21:     .ASCIZ  'CACHE DATA MEMORY CHIP SELECTION LOGIC TEST FAILED.'
11920  072216  040504  040524  046440
11921  072224  046505  051117  020131
11922  072232  044103  050111  051440
11923  072240  046105  041505  044524
11924  072246  047117  046040  043517
11925  072254  041511  052040  051505
11926  072262  020124  040506  046111
11927  072270  042105  000056
11928
11929  072274  042101  051104  051505  EM22:     .ASCII  'ADDRESS MULTIPLEXER TEST WAS UNABLE TO FORCE'
11930  072302  020123  052515  052114
11931  072310  050111  042514  042530
```

```
072316  020122  042524  052123
072324  053440  051501  052440
072332  040516  046102  020105
072340  047524  043040  051117
072346  042503
072350  040440  050040  051101          .ASCII  ' A PARITY ERROR, USING THE '<CRLF>
072356  052111  020131  051105
072364  047522  026122  052440
072372  044523  043516  052040
072400  042510  100040
072404  040515  047111  042524          .ASCII  'MAINTENANCE REGISTER, ON THE'
072412  040516  041516  020105
072420  042522  044507  052123
072426  051105  020054  047117
072434  052040  042510
072440  046440  044501  020116          .ASCIZ  ' MAIN MEMORY ADDRESS AND CONTROL LINES.'
072446  042515  047515  054522
072454  040440  042104  042522
072462  051523  040440  042116
072470  041440  047117  051124
072476  046117  046040  047111
072504  051505  000056

072510  042101  051104  051505  EM23:   .ASCII  'ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FAILED.'
072516  020123  052515  052114
072524  050111  042514  042530
072532  026122  040440  054115
072540  020054  050103  020125
072546  047111  052520  051524
072554  052040  051505  020124
072562  040506  046111  042105
072570     056
072571     200  051105  047522          .ASCIZ  <CRLF>'ERROR ADDRESS REGISTER NOT SET CORRECTLY.'
072576  020122  042101  051104
072604  051505  020123  042522
072612  044507  052123  051105
072620  047040  052117  051440
072626  052105  041440  051117
072634  042522  052103  054514
072642  000056

        072274                  EM24=EM22

        072510                  EM25=EM23

072644  042101  051104  051505  EM26:   .ASCII  'ADDRESS MEMORY, ADDRESS COMPARATOR TEST FAILURE.'
072652  020123  042515  047515
072660  054522  020054  042101
072666  051104  051505  020123
072674  047503  050115  051101
072702  052101  051117  052040
072710  051505  020124  040506
072716  046111  051125  027105
072724  040600  020116  042101          .ASCII  <CRLF>'AN ADDRESS WHICH SHOULD HAVE BEEN A HIT WAS'
072732  051104  051505  020123
072740  044127  041511  020110
```

```
072746  044123  052517  042114
072754  044040  053101  020105
072762  042502  047105  040440
072770  044040  052111  053440
072776  051501
073000  040440  046440  051511          .ASCIZ  ' A MISS.'
073006  027123     000

073011     101  042104  042522  EM27:   .ASCII  'ADDRESS MEMORY, ADDRESS COMPARATOR TEST FAILURE.'
073016  051523  046440  046505
073024  051117  026131  040440
073032  042104  042522  051523
073040  041440  046517  040520
073046  040522  047524  020122
073054  042524  052123  043040
073062  044501  052514  042522
073070     056
073071     200  047101  040440          .ASCII  <CRLF>'AN ADDRESS WHICH SHOULD HAVE BEEN A MISS '
073076  042104  042522  051523
073104  053440  044510  044103
073112  051440  047510  046125
073120  020104  040510  042526
073126  041040  042505  020116
073134  020101  044515  051523
073142     040
073143     127  051501  040440          .ASCIZ  'WAS A HIT.'
073150  044040  052111  000056

        072274                  EM30=EM22

073156  042101  051104  051505  EM31:   .ASCII  'ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FAILED.'
073164  020123  052515  052114
073172  050111  042514  042530
073200  026122  040440  054115
073206  020054  047125  041111
073214  051525  044440  050116
073222  052125  020123  042524
073230  052123  043040  044501
073236  042514  027104
073242  042600  051122  051117          .ASCIZ  <CRLF>'ERROR ADDRESS REGISTER NOT SET CORRECTLY.'
073250  040440  042104  042522
073256  051523  051040  043505
073264  051511  042524  020122
073272  047516  020124  042523
073300  020124  047503  051122
073306  041505  046124  027131
073314     000

        072274                  EM32=EM22

        073156                  EM33=EM31

073315     101  042104  042522  EM34:   .ASCII  'ADDRESS MULTIPLEXER, AMX, DUAL ADDRESS TEST,'<CRLF>
073322  051523  046440  046125
073330  044524  046120  054105
073336  051105  020054  046501
```

```
 12044  073344  026130  042040  040525
 12045  073352  020114  042101  051104
 12046  073360  051505  020123  042524
 12047  073366  052123  100054
 12048  073372  047117  041440  052520          .ASCIZ  'ON CPU INPUTS, FAILED.'
 12049  073400  044440  050116  052125
 12050  073406  026123  043040  044501
 12051  073414  042514  027104     000
 12052
 12053  073421     101  042104  042522  EM35:   .ASCII  'ADDRESS MULTIPLEXER, AMX, DUAL ADDRESS TEST,'<CRLF>
 12054  073426  051523  046440  046125
 12055  073434  044524  046120  054105
 12056  073442  051105  020054  046501
 12057  073450  026130  042040  040525
 12058  073456  020114  042101  051104
 12059  073464  051505  020123  042524
 12060  073472  052123  100054
 12061  073476  047117  052440  044516          .ASCIZ  'ON UNIBUS INPUTS, FAILED.'
 12062  073504  052502  020123  047111
 12063  073512  052520  051524  020054
 12064  073520  040506  046111  042105
 12065  073526  000056
 12066
 12067  073530  042101  051104  051505  EM36:   .ASCII  'ADDRESS MEMORY COUNT PATTERN TEST FAILURE,'<CRLF>
 12068  073536  020123  042515  047515
 12069  073544  054522  041440  052517
 12070  073552  052116  050040  052101
 12071  073560  042524  047122  052040
 12072  073566  051505  020124  040506
 12073  073574  046111  051125  026105
 12074  073602     200
 12075  073603     116  020117  040520          .ASCIZ  'NO PARITY ERROR OCCURS, BUT CAN NOT GET A HIT.'
 12076  073610  044522  054524  042440
 12077  073616  051122  051117  047440
 12078  073624  041503  051125  026123
 12079  073632  041040  052125  041440
 12080  073640  047101  047040  052117
 12081  073646  043440  052105  040440
 12082  073654  044040  052111  000056
 12083
 12084  073662  042101  051104  051505  EM37:   .ASCIZ  'ADDRESS MEMORY COUNT PATTERN TEST, ERROR SUMMARY.'
 12085  073670  020123  042515  047515
 12086  073676  054522  041440  052517
 12087  073704  052116  050040  052101
 12088  073712  042524  047122  052040
 12089  073720  051505  026124  042440
 12090  073726  051122  051117  051440
 12091  073734  046525  040515  054522
 12092  073742  000056
 12093
 12094  073744  042101  051104  051505  EM40:   .ASCII  'ADDRESS MEMORY COUNT PATTERN TEST FAILURE,'<CRLF>
 12095  073752  020123  042515  047515
 12096  073760  054522  041440  052517
 12097  073766  052116  050040  052101
 12098  073774  042524  047122  052040
 12099  074002  051505  020124  040506
```

```
074010  046111  051125  026105
074016     200
074017     103  041501  042510          .ASCII  'CACHE MEMORY ADDRESS PARITY ERROR OCCURRED'
074024  046440  046505  051117
074032  020131  042101  051104
074040  051505  020123  040520
074046  044522  054524  042440
074054  051122  051117  047440
074062  041503  051125  042522
074070     104
074071     040  052101  052040          .ASCIZ  ' AT THE TEST ADDRESS.'
074076  042510  052040  051505
074104  020124  042101  051104
074112  051505  027123     000

074117     101  042104  042522  EM41:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILED TO GET '
074124  051523  046440  046505
074132  051117  020131  052504
074140  046101  040440  042104
074146  042522  051523  052040
074154  051505  020124  040506
074162  046111  042105  052040
074170  020117  042507  020124
074176  020101  044510  020124          .ASCII  'A HIT AT A TEST ADDRESS,'<CRLF>
074204  052101  040440  052040
074212  051505  020124  042101
074220  051104  051505  026123
074226     200
074227     127  044510  042514          .ASCIZ  'WHILE WRITING THE ADDRESS MEMORY LOCATIONS.'
074234  053440  044522  044524
074242  043516  052040  042510
074250  040440  042104  042522
074256  051523  046440  046505
074264  051117  020131  047514
074272  040503  044524  047117
074300  027123     000

074303     101  042104  042522  EM42:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILED TO GET'
074310  051523  046440  046505
074316  051117  020131  052504
074324  046101  040440  042104
074332  042522  051523  052040
074340  051505  020124  040506
074346  046111  042105  052040
074354  020117  042507     124
074361     101  044040  052111          .ASCII  'A HIT AT A TEST ADDRESS,'<CRLF>
074366  040440  020124  020101
074374  042524  052123  040440
074402  042104  042522  051523
074410  100054
```

```
074412  044127  046111  020105          .ASCII  'WHILE READING BACK THE ADDRESS MEMORY LOCATIONS.'<CRLF><LF>
074420  042522  042101  047111
074426  020107  040502  045503
074434  052040  042510  040440
074442  042104  042522  051523
074450  046440  046505  051117
074456  020131  047514  040503
074464  044524  047117  027123
074472  005200
074474  052133  044510  020123          .ASCIZ  '[THIS PROBLEM MIGHT BE CORRECTED BY ECO M8182-4]'<CRLF>
074502  051120  041117  042514
074510  020115  044515  044107
074516  020124  042502  041440
074524  051117  042522  052103
074532  042105  041040  020131
074540  041505  020117  034115
074546  034061  026462  056464
074554  000200
```

```
12168
```

```
 12169  074556  042101  051104  051505  EM43:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILURE,'<CRLF>
 12170  074564  020123  042515  047515
 12171  074572  054522  042040  040525
 12172  074600  020114  042101  051104
 12173  074606  051505  020123  042524
 12174  074614  052123  043040  044501
 12175  074622  052514  042522  100054
 12176  074630  040503  044103  020105          .ASCIZ  'CACHE ADDRESS MEMORY PARITY ERROR OCCURRED.'
 12177  074636  042101  051104  051505
 12178  074644  020123  042515  047515
 12179  074652  054522  050040  051101
 12180  074660  052111  020131  051105
 12181  074666  047522  020122  041517
 12182  074674  052503  051122  042105
 12183  074702  000056
```

```

074704  040515  047111  046440  EM44:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED,'
074712  046505  051117  020131
074720  054502  042524  046440
074726  051501  020113  042507
074734  042516  040522  047524
074742  020122  042524  052123
074750  043040  044501  042514
074756  026104
074760  042040  044517  043516          .ASCII  ' DOING CPU DATOB.'<CRLF>
074766  041440  052520  042040
074774  052101  041117  100056
075002  020101  040515  047111          .ASCII  'A MAIN MEMORY ADDRESS AND CONTROL LINE '
075010  046440  046505  051117
075016  020131  042101  051104
075024  051505  020123  047101
075032  020104  047503  052116
075040  047522  020114  044514
075046  042516     040
075051     120  051101  052111          .ASCIZ  'PARITY ERROR OCCURRED.'
075056  020131  051105  047522
075064  020122  041517  052503
075072  051122  042105  000056

075100  040515  047111  046440  EM45:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED,'
075106  046505  051117  020131
075114  054502  042524  046440
075122  051501  020113  042507
075130  042516  040522  047524
075136  020122  042524  052123
075144  043040  044501  042514
075152  026104
075154  042040  044517  043516          .ASCII  ' DOING UNIBUS DATOB.'<CRLF>
075162  052440  044516  052502
075170  020123  040504  047524
075176  027102     200
075201     101  046440  044501          .ASCII  'A MAIN MEMORY ADDRESS AND CONTROL LINE '
075206  020116  042515  047515
075214  054522  040440  042104
075222  042522  051523  040440
075230  042116  041440  047117
075236  051124  046117  046040
075244  047111  020105
075250  040520  044522  054524          .ASCIZ  'PARITY ERROR OCCURRED.'
075256  042440  051122  051117
075264  047440  041503  051125
075272  042522  027104     000

075277     115  044501  020116  EM46:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED.'
075304  042515  047515  054522
075312  041040  052131  020105
075320  040515  045523  043440
075326  047105  051105  052101
075334  051117  052040  051505
075342  020124  040506  046111
075350  042105     056
```

```
 12240  075353     200  051127  047117             .ASCIZ  <CRLF>'WRONG BYTE WRITTEN, ON A CPU DATOB.'
 12241  075360  020107  054502  042524
 12242  075366  053440  044522  052124
 12243  075374  047105  020054  047117
 12244  075402  040440  041440  052520
 12245  075410  042040  052101  041117
 12246  075416  000056
 12247
 12248  075420  040515  047111  046440  EM47:     .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED.'
 12249  075426  046505  051117  020131
 12250  075434  054502  042524  046440
 12251  075442  051501  020113  042507
 12252  075450  042516  040522  047524
 12253  075456  020122  042524  052123
 12254  075464  043040  044501  042514
 12255  075472  027104
 12256  075474  053600  047522  043516            .ASCIZ  <CRLF>'WRONG BYTE WRITTEN, ON A UNIBUS DATOB.'
 12257  075502  041040  052131  020105
 12258  075510  051127  052111  042524
 12259  075516  026116  047440  020116
 12260  075524  020101  047125  041111
 12261  075532  051525  042040  052101
 12262  075540  041117  000056
 12263
 12264          074704                  EM50=EM44
 12265
 12266          075100                  EM51=EM45
 12267
 12268          075277                  EM52=EM46
 12269
 12270          075420                  EM53=EM47
 12271
 12272  075544  040503  044103  020105  EM54:     .ASCII  'CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST FAILED.'
 12273  075552  042101  051104  051505
 12274  075560  020123  042515  047515
 12275  075566  054522  050040  053517
 12276  075574  051105  052440  020120
 12277  075602  047111  040526  044514
 12278  075610  040504  047524  020122
 12279  075616  042524  052123  043040
 12280  075624  044501  042514  027104
 12281  075632  041600  041501  042510            .ASCII  <CRLF>'CACHE DATA OR ADDRESS MEMORY PARITY '
 12282  075640  042040  052101  020101
 12283  075646  051117  040440  042104
 12284  075654  042522  051523  046440
 12285  075662  046505  051117  020131
 12286  075670  040520  044522  054524
 12287  075676     040
 12288  075677     105  051122  051117            .ASCIZ  'ERROR DETECTED.'
 12289  075704  042040  052105  041505
 12290  075712  042524  027104     000
 12291  075717     103  051103  041440  EM55:     .ASCIZ  /CCR COULD NOT BE CLEARED/
 12292  075724  052517  042114  047040
 12293  075732  052117  041040  020105
 12294  075740  046103  040505  042522
 12295  075746  000104
```

```
075750  053111  051523  024040   EM56:  .ASCIZ  /IVSS (BIT 14) COULD NOT BE SET IN CCR/
075756  044502  020124  032061
075764  020051  047503  046125
075772  020104  047516  020124
076000  042502  051440  052105
076006  044440  020116  041503
076014  000122
076016  053111  051523  041440   EM57:  .ASCIZ  /IVSS COULD NOT BE CLEARED IN CCR/
076024  052517  042114  047040
076032  052117  041040  020105
076040  046103  040505  042522
076046  020104  047111  041440
076054  051103     000
076057     126  044523  020125   EM60:  .ASCIZ  /VSIU (BIT 13) COULD NOT BE SET/
076064  041050  052111  030440
076072  024463  041440  052517
076100  042114  047040  052117
076106  041040  020105  042523
076114  000124
076116  041526  050111  042040   EM61:  .ASCIZ  /VCIP DID NOT CLEAR AFTER CACHE FLUSH (ON SETTING VSIU)/
076124  042111  047040  052117
076132  041440  042514  051101
076140  040440  052106  051105
076146  041440  041501  042510
076154  043040  052514  044123
076162  024040  047117  051440
076170  052105  044524  043516
076176  053040  044523  024525
076204     000
076205     126  044523  020125   EM62:  .ASCIZ  /VSIU COULD NOT BE CLEARED/
076212  047503  046125  020104
076220  047516  020124  042502
076226  041440  042514  051101
076234  042105     000
076237     126  044503  020120   EM63:  .ASCIZ  /VCIP DID NOT SET WHEN CACHE FLUSH BIT WAS SET/
076244  044504  020104  047516
076252  020124  042523  020124
076260  044127  047105  041440
076266  041501  042510  043040
076274  052514  044123  041040
076302  052111  053440  051501
076310  051440  052105     000
076315     126  044523  020125   EM64:  .ASCIZ  /VSIU DID NOT SWITCH WHEN CACHE FLUSH BIT WAS SET/
076322  044504  020104  047516
076330  020124  053523  052111
076336  044103  053440  042510
076344  020116  040503  044103
076352  020105  046106  051525
076360  020110  044502  020124
076366  040527  020123  042523
076374  000124
076376  051526  052511  051440   EM65:  .ASCIZ  /VSIU SWITCHED WHEN CACHE FLUSH WAS DONE,WITH IVSS SET/
076404  044527  041524  042510
076412  020104  044127  047105
076420  041440  041501  042510
076426  043040  052514  044123
```

```
076434  053440  051501  042040
076442  047117  026105  044527
076450  044124  044440  051526
076456  020123  042523  000124
076464  042524  052123  042055   EM66:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
076472  052101  020101  042522
076500  042506  042522  041516
076506  020105  047516  020124
076514  020101  044515  051523
076522  005015  040526  044514           .ASCIZ  <15><12>/VALID STORE NOT SWITCHED ON CACHE FLUSH/
076530  020104  052123  051117
076536  020105  047516  020124
076544  053523  052111  044103
076552  042105  047440  020116
076560  040503  044103  020105
076566  046106  051525  000110
076574  042524  052123  042055   EM67:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
076602  052101  020101  042522
076610  042506  042522  041516
076616  020105  047516  020124
076624  020101  044515  051523
076632  005015  040526  044514           .ASCIZ  <15><12>/VALID STORE NOT INVALIDATED ON CACHE FLUSH/
076640  020104  052123  051117
076646  020105  047516  020124
076654  047111  040526  044514
076662  040504  042524  020104
076670  047117  041440  041501
076676  042510  043040  052514
076704  044123     000
076707     124  051505  026524   EM70:  .ASCII  /TEST-DATA REFERENCE NOT A HIT/
076714  040504  040524  051040
076722  043105  051105  047105
076730  042503  047040  052117
076736  040440  044040  052111
076744  005015  051106  046517           .ASCIZ  <15><12>/FROM THE GROUP AND VALID STORE BEING CHECKED/
076752  052040  042510  043440
076760  047522  050125  040440
076766  042116  053040  046101
076774  042111  051440  047524
077002  042522  041040  044505
077010  043516  041440  042510
077016  045503  042105     000
077023     104  052101  020101   EM71:  .ASCIZ  /DATA ERROR ON READING CACHED LOCATION/
077030  051105  047522  020122
077036  047117  051040  040505
077044  044504  043516  041440
077052  041501  042510  020104
077060  047514  040503  044524
077066  047117     000
077071     124  051505  026524   EM72:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
077076  040504  040524  051040
077104  043105  051105  047105
077112  042503  047040  052117
077120  040440  046440  051511
077126     123
077127     015  041412  041501           .ASCIZ  <15><12>/CACHE DOES NOT TURN OFF, WHEN FLUSH DONE WITH IVSS SET/
```

```
077134  042510  042040  042517
077142  020123  047516  020124
077150  052524  047122  047440
077156  043106  020054  044127
077164  047105  043040  052514
077172  044123  042040  047117
077200  020105  044527  044124
077206  044440  051526  020123
077214  042523  000124
077220  042524  052123  042055   EM73:  .ASCII  /TEST-DATA REFERENCE NOT A HIT/
077226  052101  020101  042522
077234  042506  042522  041516
077242  020105  047516  020124
077250  020101  044510     124
077255     015  041412  041501          .ASCIZ  <15><12>/CACHE DOES NOT TURN ON AFTER TURNING OFF/
077262  042510  042040  042517
077270  020123  047516  020124
077276  052524  047122  047440
077304  020116  043101  042524
077312  020122  052524  047122
077320  047111  020107  043117
077326  000106
077330  042524  052123  042055   EM74:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
077336  052101  020101  042522
077344  042506  042522  041516
077352  020105  047516  020124
077360  020101  044515  051523
077366  005015  040503  044103          .ASCIZ  <15><12>/CACHE BYPASS DID NOT FORCE A MISS/
077374  020105  054502  040520
077402  051523  042040  042111
077410  047040  052117  043040
077416  051117  042503  040440
077424  046440  051511  000123
077432  042524  052123  042055   EM75:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
077440  052101  020101  042522
077446  042506  042522  041516
077454  020105  047516  020124
077462  020101  044515  051523
077470  005015  040503  044103          .ASCIZ  <15><12>/CACHE BYPASS DID NOT INVALIDATE CACHED DATA/
077476  020105  054502  040520
077504  051523  042040  042111
077512  047040  052117  044440
077520  053116  046101  042111
077526  052101  020105  040503
077534  044103  042105  042040
077542  052101  000101
077546  042524  052123  042055   EM76:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
077554  052101  020101  042522
077562  042506  042522  041516
077570  020105  047516  020124
077576  020101  044515  051523
077604  005015  051501  041122          .ASCIZ  <15><12>/ASRB DID NOT FORCE A MISS ON THE OPERAND/
077612  042040  042111  047040
077620  052117  043040  051117
077626  042503  040440  046440
077634  051511  020123  047117
```

```
12464  077642  052040  042510  047440
12465  077650  042520  040522  042116
12466  077656     000
12467  077657     124  051505  026524  EM77:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
12468  077664  040504  040524  051040
12469  077672  043105  051105  047105
12470  077700  042503  047040  052117
12471  077706  040440  046440  051511
12472  077714     123
12473  077715     015  041412  041501          .ASCII  <15><12>/CACHED OPERAND NOT INVALIDATED ON ASRB EXECUTION/<CRLF>
12474  077722  042510  020104  050117
12475  077730  051105  047101  020104
12476  077736  047516  020124  047111
12477  077744  040526  044514  040504
12478  077752  042524  020104  047117
12479  077760  040440  051123  020102
12480  077766  054105  041505  052125
12481  077774  047511  100116
12482  100000  052133  044510  020123          .ASCIZ  /[THIS PROBLEM MIGHT BE CORRECTED BY ECO M8182-4]/<CRLF>
12483  100006  051120  041117  042514
12484  100014  020115  044515  044107
12485  100022  020124  042502  041440
12486  100030  051117  042522  052103
12487  100036  042105  041040  020131
12488  100044  041505  020117  034115
12489  100052  034061  026462  056464
12490  100060  000200
12491  100062  042524  052123  042055  EM100: .ASCIZ  /TEST-DATA COULD NOT BE MADE HIT/
12492  100070  052101  020101  047503
12493  100076  046125  020104  047516
12494  100104  020124  042502  046440
12495  100112  042101  020105  044510
12496  100120  000124
12497  100122  047516  050040  051101  EM103: .ASCIZ  /NO PARITY ERROR TRAP ON VALID STORE PARITY ERROR/
12498  100130  052111  020131  051105
12499  100136  047522  020122  051124
12500  100144  050101  047440  020116
12501  100152  040526  044514  020104
12502  100160  052123  051117  020105
12503  100166  040520  044522  054524
12504  100174  042440  051122  051117
12505  100202     000
12506  100203     124  051505  026524  EM104: .ASCII  /TEST-DATA-REFERENCE GIVING VALID STORE PARITY/
12507  100210  040504  040524  051055
12508  100216  043105  051105  047105
12509  100224  042503  043440  053111
12510  100232  047111  020107  040526
12511  100240  044514  020104  052123
12512  100246  051117  020105  040520
12513  100254  044522  054524
12514  100260  005015  051105  047522          .ASCIZ  <15><12>/ERROR WAS NOT A MISS/
12515  100266  020122  040527  020123
12516  100274  047516  020124  020101
12517  100302  044515  051523     000
12518  100307     106  050126  020105  EM105: .ASCIZ  /FVPE DID NOT GET CLEARED AFTER VSPE OCCURED/
12519  100314  044504  020104  047516
```

```
 12520  100322  020124  042507  020124
 12521  100330  046103  040505  042522
 12522  100336  020104  043101  042524
 12523  100344  020122  051526  042520
 12524  100352  047440  041503  051125
 12525  100360  042105     000
 12526  100363     126  046101  042111   EM106: .ASCIZ  /VALID-STORE-PARITY-ERROR BIT DID NOT SET IN CCR ON VSPE/
 12527  100370  051455  047524  042522
 12528  100376  050055  051101  052111
 12529  100404  026531  051105  047522
 12530  100412  020122  044502  020124
 12531  100420  044504  020104  047516
 12532  100426  020124  042523  020124
 12533  100434  047111  041440  051103
 12534  100442  047440  020116  051526
 12535  100450  042520     000
 12536  100453     106  051501  020124   EM107: .ASCII  /FAST ADDRESS MEMORY PARITY ERROR BITS (4,5) NOT/
 12537  100460  042101  051104  051505
 12538  100466  020123  042515  047515
 12539  100474  054522  050040  051101
 12540  100502  052111  020131  051105
 12541  100510  047522  020122  044502
 12542  100516  051524  024040  026064
 12543  100524  024465  047040  052117
 12544  100532  005015  042523  020124          .ASCIZ  <15><12>/SET CORRECTLY IN MSER ON VSPE/
 12545  100540  047503  051122  041505
 12546  100546  046124  020131  047111
 12547  100554  046440  042523  020122
 12548  100562  047117  053040  050123
 12549  100570  000105
 12550  100572  051526  052511  051440   EM110: .ASCIZ  /VSIU SWITCHED ON VSPE/
 12551  100600  044527  041524  042510
 12552  100606  020104  047117  053040
 12553  100614  050123  000105
 12554  100620  042515  047515  054522  EM111:  .ASCIZ  /MEMORY SYSTEM ERROR REGISTER COULD NOT BE CLEARED/
 12555  100626  051440  051531  042524
 12556  100634  020115  051105  047522
 12557  100642  020122  042522  044507
 12558  100650  052123  051105  041440
 12559  100656  052517  042114  047040
 12560  100664  052117  041040  020105
 12561  100672  046103  040505  042522
 12562  100700  000104
 12563  100702  051526  042520  041440   EM112: .ASCIZ  /VSPE COULD NOT BE CLEARED IN CCR/
 12564  100710  052517  042114  047040
 12565  100716  052117  041040  020105
 12566  100724  046103  040505  042522
 12567  100732  020104  047111  041440
 12568  100740  051103     000
 12569  100743     124  051505  026524   EM113: .ASCIZ  /TEST-DATA-REFERENCE NOT A HIT/
 12570  100750  040504  040524  051055
 12571  100756  043105  051105  047105
 12572  100764  042503  047040  052117
 12573  100772  040440  044040  052111
 12574  101000     000
 12575  101001     124  051505  026524  EM115:  .ASCII  /TEST-DATA-REFERNECE NOT A MISS/
```

```
101006  040504  040524  051055
101014  043105  051105  042516
101022  042503  047040  052117
101030  040440  046440  051511
101036     123
101037     015  041412  041501                .ASCIZ  <15><12>/CACHE DID NOT TURN OFF ON BACK-TO-BACK FLUSH/
101044  042510  042040  042111
101052  047040  052117  052040
101060  051125  020116  043117
101066  020106  047117  041040
101074  041501  026513  047524
101102  041055  041501  020113
101110  046106  051525  000110

101116  054502  020120  044502  EM123:  .ASCIZ  ?BYP BIT IN KIPDR COULD NOT BE CLEARED?
101124  020124  047111  045440
101132  050111  051104  041440
101140  052517  042114  047040
101146  052117  041040  020105
101154  046103  040505  042522
101162  000104
101164  054502  020120  044502  EM124:  .ASCIZ  ?BYP BIT IN KIPDR COULD NOT BE SET?
101172  020124  047111  045440
101200  050111  051104  041440
101206  052517  042114  047040
101214  052117  041040  020105
101222  042523  000124
101226  042524  052123  042055  EM125:  .ASCIZ  /TEST-DATA COULD NOT BE MADE HIT/
101234  052101  020101  047503
101242  046125  020104  047516
101250  020124  042502  046440
101256  042101  020105  044510
101264  000124
101266  042524  052123  042055   EM126: .ASCII  /TEST-DATA REFERENCE NOT A MISS/
101274  052101  020101  042522
101302  042506  042522  041516
101310  020105  047516  020124
101316  020101  044515  051523
101324  005015  040503  044103          .ASCIZ  <15><12>/CACHED DATA WAS NOT FORCED A MISS ON VIRTUAL PAGE BYPASS/
101332  042105  042040  052101
101340  020101  040527  020123
101346  047516  020124  047506
101354  041522  042105  040440
101362  046440  051511  020123
101370  047117  053040  051111
101376  052524  046101  050040
101404  043501  020105  054502
101412  040520  051523     000
101417     124  051505  020124   EM127: .ASCII  /TEST DATA REFERENCE NOT A MISS/
101424  040504  040524  051040
101432  043105  051105  047105
101440  042503  047040  052117
101446  040440  046440  051511
101454     123
101455     015  041412  041501          .ASCIZ  <15><12>/CACHED DATA WAS NOT INVALIDATED ON VIRTUAL PAGE BYPASS/
101462  042510  020104  040504
```

```
101470  040524  053440  051501
101476  047040  052117  044440
101504  053116  046101  042111
101512  052101  042105  047440
101520  020116  044526  052122
101526  040525  020114  040520
101534  042507  041040  050131
101542  051501  000123
101546  054502  020120  044502  EM130:  .ASCIZ  ?BYP BIT IN SIPDR COULD NOT BE CLEARED?
101554  020124  047111  051440
101562  050111  051104  041440
101570  052517  042114  047040
101576  052117  041040  020105
101604  046103  040505  042522
101612  000104
101614  054502  020120  044502  EM131:  .ASCIZ  ?BYP BIT IN SIPDR COULD NOT BE SET?
101622  020124  047111  051440
101630  050111  051104  041440
101636  052517  042114  047040
101644  052117  041040  020105
101652  042523  000124
101656  054502  020120  044502  EM132:  .ASCIZ  ?BYP BIT IN UIPDR COULD NOT BE CLEARED?
101664  020124  047111  052440
101672  050111  051104  041440
101700  052517  042114  047040
101706  052117  041040  020105
101714  046103  040505  042522
101722  000104
101724  054502  020120  044502  EM133:  .ASCIZ  ?BYP BIT IN UIPDR COULD NOT BE SET?
101732  020124  047111  052440
101740  050111  051104  041440
101746  052517  042114  047040
101754  052117  041040  020105
101762  042523  000124
101766  040503  044103  020105  EM136:  .ASCII  'CACHE ADDRESS MEMORY PARITY LOGIC TEST FAILED.'<CRLF>
101774  042101  051104  051505
102002  020123  042515  047515
102010  054522  050040  051101
102016  052111  020131  047514
102024  044507  020103  042524
102032  052123  043040  044501
102040  042514  027104     200
102045     125  040516  046102          .ASCII  'UNABLE TO FORCE A PARITY ERROR ON THE LOW BYTE '
102052  020105  047524  043040
102060  051117  042503  040440
102066  050040  051101  052111
102074  020131  051105  047522
102102  020122  047117  052040
102110  042510  046040  053517
102116  041040  052131  020105
102124  043117  040440  020116          .ASCIZ  'OF AN ADDRESS,'<CRLF>'USING THE MAINTENANCE REGISTER.'
102132  042101  051104  051505
102140  026123  052600  044523
102146  043516  052040  042510
102154  046440  044501  052116
102162  047105  047101  042503
```

```
 12688  102170  051040  043505  051511
 12689  102176  042524  027122     000
 12690
 12691  102203     103  041501  042510  EM137:  .ASCII  'CACHE ADDRESS MEMORY PARITY LOGIC TEST FAILED.'
 12692  102210  040440  042104  042522
 12693  102216  051523  046440  046505
 12694  102224  051117  020131  040520
 12695  102232  044522  054524  046040
 12696  102240  043517  041511  052040
 12697  102246  051505  020124  040506
 12698  102254  046111  042105     056
 12699  102261     200  047125  041101          .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR ON THE HIGH BYTE '
 12700  102266  042514  052040  020117
 12701  102274  047506  041522  020105
 12702  102302  020101  040520  044522
 12703  102310  054524  042440  051122
 12704  102316  051117  047440  020116
 12705  102324  044124  020105  044510
 12706  102332  044107  041040  052131
 12707  102340  020105
 12708  102342  043117  040440  020116          .ASCIZ  'OF AN ADDRESS.'<CRLF>'USING THE MAINTENANCE REGISTER.'
 12709  102350  042101  051104  051505
 12710  102356  026123  052600  044523
 12711  102364  043516  052040  042510
 12712  102372  046440  044501  052116
 12713  102400  047105  047101  042503
 12714  102406  051040  043505  051511
 12715  102414  042524  027122     000
 12716
 12717  102421                          EM140:
 12718  102421     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 12719  102426  042515  047515  054522
 12720  102434  042040  052101  020101
 12721  102442  040520  044522  054524
 12722  102450  041440  042510  045503
 12723  102456  051105  020123  042524
 12724  102464  052123  043040  044501
 12725  102472  042514  027104
 12726  102476  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 12727  102504  020105  047524  043040
 12728  102512  051117  042503  040440
 12729  102520  050040  051101  052111
 12730  102526  020131  051105  047522
 12731  102534  026122  052440  044523
 12732  102542  043516     040
 12733  102545     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 12734  102552  044501  052116  047105
 12735  102560  047101  042503  051040
 12736  102566  043505  051511  042524
 12737  102574  026122     200
 12738  102577     101  020124  044124          .ASCII  'AT THE MAIN MEMORY EVEN WORD, LOW BYTE, PARITY '
 12739  102604  020105  040515  047111
 12740  102612  046440  046505  051117
 12741  102620  020131  053105  047105
 12742  102626  053440  051117  026104
 12743  102634  046040  053517  041040
```

```
102642  052131  026105  050040
102650  051101  052111  020131
102656  044103  041505  042513              .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
102664  026122  020200  042522
102672  042101  047111  020107
102700  020101  040504  040524
102706  050040  052101  042524
102714  047122  053440  044510
102722  044103     040
102725     123  047510  046125              .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
102732  020104  040510  042526
102740  041440  052501  042523
102746  020104  047101  042440
102754  051122  051117  000056

102762                              EM141:
102762  040515  047111  046440              .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
102770  046505  051117  020131
102776  040504  040524  050040
103004  051101  052111  020131
103012  044103  041505  042513
103020  051522  052040  051505
103026  020124  040506  046111
103034  042105     056
103037     200  047125  041101              .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
103044  042514  052040  020117
103052  047506  041522  020105
103060  020101  040520  044522
103066  054524  042440  051122
103074  051117  020054  051525
103102  047111  020107
103106  044124  020105  040515              .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
103114  047111  042524  040516
103122  041516  020105  042522
103130  044507  052123  051105
103136  100054
103140  052101  052040  042510              .ASCII  'AT THE MAIN MEMORY ODD WORD, LOW BYTE, PARITY '
103146  046440  044501  020116
103154  042515  047515  054522
103162  047440  042104  053440
103170  051117  026104  046040
103176  053517  041040  052131
103204  026105  050040  051101
103212  052111  020131
103216  044103  041505  042513              .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
103224  026122  020200  042522
103232  042101  047111  020107
103240  020101  040504  040524
103246  050040  052101  042524
103254  047122  053440  044510
103262  044103     040
103265     123  047510  046125              .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
103272  020104  040510  042526
103300  041440  052501  042523
103306  020104  047101  042440
103314  051122  051117  000056
```

```

103322                          EM142:
103322  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
103330  046505  051117  020131
103336  040504  040524  050040
103344  051101  052111  020131
103352  044103  041505  042513
103360  051522  052040  051505
103366  020124  040506  046111
103374  042105     056
103377     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
103404  042514  052040  020117
103412  047506  041522  020105
103420  020101  040520  044522
103426  054524  042440  051122
103434  051117  020054  051525
103442  047111  020107
103446  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
103454  047111  042524  040516
103462  041516  020105  042522
103470  044507  052123  051105
103476  100054
103500  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY EVEN WORD, HIGH BYTE, PARITY '
103506  046440  044501  020116
103514  042515  047515  054522
103522  042440  042526  020116
103530  047527  042122  020054
103536  044510  044107  041040
103544  052131  026105  050040
103552  051101  052111  020131
103560  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
103566  026122  020200  042522
103574  042101  047111  020107
103602  020101  040504  040524
103610  050040  052101  042524
103616  047122  053440  044510
103624  044103     040
103627     123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
103634  020104  040510  042526
103642  041440  052501  042523
103650  020104  047101  042440
103656  051122  051117  000056

103664                          EM143:
103664  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
103672  046505  051117  020131
103700  040504  040524  050040
103706  051101  052111  020131
103714  044103  041505  042513
103722  051522  052040  051505
103730  020124  040506  046111
103736  042105     056
103741     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
103746  042514  052040  020117
103754  047506  041522  020105
103762  020101  040520  044522
```

```
 12856  103770  054524  042440  051122
 12857  103776  051117  020054  051525
 12858  104004  047111  020107
 12859  104010  044124  020105  040515           .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 12860  104016  047111  042524  040516
 12861  104024  041516  020105  042522
 12862  104032  044507  052123  051105
 12863  104040  100054
 12864  104042  052101  052040  042510           .ASCII  'AT THE MAIN MEMORY ODD WORD, HIGH BYTE, PARITY '
 12865  104050  046440  044501  020116
 12866  104056  042515  047515  054522
 12867  104064  047440  042104  053440
 12868  104072  051117  026104  044040
 12869  104100  043511  020110  054502
 12870  104106  042524  020054  040520
 12871  104114  044522  054524     040
 12872  104121     103  042510  045503           .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 12873  104126  051105  100054  051040
 12874  104134  040505  044504  043516
 12875  104142  040440  042040  052101
 12876  104150  020101  040520  052124
 12877  104156  051105  020116  044127
 12878  104164  041511  020110
 12879  104170  044123  052517  042114           .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 12880  104176  044040  053101  020105
 12881  104204  040503  051525  042105
 12882  104212  040440  020116  051105
 12883  104220  047522  027122     000
 12884
 12885  104225                           EM144:
 12886  104225     103  041501  042510           .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
 12887  104232  042040  052101  020101
 12888  104240  042515  047515  054522
 12889  104246  050040  051101  052111
 12890  104254  020131  044103  041505
 12891  104262  042513  051522  052040
 12892  104270  051505  020124  040506
 12893  104276  046111  042105     056
 12894  104303     200  047125  041101           .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
 12895  104310  042514  052040  020117
 12896  104316  047506  041522  020105
 12897  104324  020101  040520  044522
 12898  104332  054524  042440  051122
 12899  104340  051117  020054  051525
 12900  104346  047111  020107
 12901  104352  044124  020105  040515           .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 12902  104360  047111  042524  040516
 12903  104366  041516  020105  042522
 12904  104374  044507  052123  051105
 12905  104402  100054
 12906  104404  052101  052040  042510           .ASCII  'AT THE GROUP ZERO,LOW BYTE, DATA PARITY CHECKER,'
 12907  104412  043440  047522  050125
 12908  104420  055040  051105  026117
 12909  104426  047514  020127  054502
 12910  104434  042524  020054  040504
 12911  104442  040524  050040  051101
```

```
12912  104450  052111  020131  044103
12913  104456  041505  042513  026122
12914  104464  051200  040505  044504            .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
12915  104472  043516  040440  042040
12916  104500  052101  020101  040520
12917  104506  052124  051105  020116
12918  104514  044127  041511  020110
12919  104522  044123  052517  042114
12920  104530  044040  053101  020105
12921  104536  040503  051525  042105            .ASCIZ  'CAUSED AN ERROR.'
12922  104544  040440  020116  051105
12923  104552  047522  027122     000
12924
12925  104557                            EM145:
12926  104557     103  041501  042510            .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
12927  104564  042040  052101  020101
12928  104572  042515  047515  054522
12929  104600  050040  051101  052111
12930  104606  020131  044103  041505
12931  104614  042513  051522  052040
12932  104622  051505  020124  040506
12933  104630  046111  042105     056
12934  104635     200  047125  041101            .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
12935  104642  042514  052040  020117
12936  104650  047506  041522  020105
12937  104656  020101  040520  044522
12938  104664  054524  042440  051122
12939  104672  051117  020054  051525
12940  104700  047111  020107
12941  104704  044124  020105  040515            .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
12942  104712  047111  042524  040516
12943  104720  041516  020105  042522
12944  104726  044507  052123  051105
12945  104734  100054
12946  104736  052101  052040  042510            .ASCII  'AT THE GROUP ONE,LOW BYTE, DATA PARITY CHECKER,'
12947  104744  043440  047522  050125
12948  104752  047440  042516  046054
12949  104760  053517  041040  052131
12950  104766  026105  042040  052101
12951  104774  020101  040520  044522
12952  105002  054524  041440  042510
12953  105010  045503  051105     054
12954  105015     200  042522  042101            .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
12955  105022  047111  020107  020101
12956  105030  040504  040524  050040
12957  105036  052101  042524  047122
12958  105044  053440  044510  044103
12959  105052  051440  047510  046125
12960  105060  020104  040510  042526
12961  105066     040
12962  105067     103  052501  042523            .ASCIZ  'CAUSED AN ERROR.'
12963  105074  020104  047101  042440
12964  105102  051122  051117  000056
12965
12966  105110                            EM146:
12967  105110  040503  044103  020105            .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
```

```
 12968  105116  040504  040524  046440
 12969  105124  046505  051117  020131
 12970  105132  040520  044522  054524
 12971  105140  041440  042510  045503
 12972  105146  051105  020123  042524
 12973  105154  052123  043040  044501
 12974  105162  042514  027104
 12975  105166  052600  040516  046102            .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
 12976  105174  020105  047524  043040
 12977  105202  051117  042503  040440
 12978  105210  050040  051101  052111
 12979  105216  020131  051105  047522
 12980  105224  026122  052440  044523
 12981  105232  043516     040
 12982  105235     124  042510  046440            .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 12983  105242  044501  052116  047105
 12984  105250  047101  042503  051040
 12985  105256  043505  051511  042524
 12986  105264  026122     200
 12987  105267     101  020124  044124            .ASCII  'AT THE GROUP ZERO,HIGH BYTE, DATA PARITY CHECKER,'
 12988  105274  020105  051107  052517
 12989  105302  020120  042532  047522
 12990  105310  044054  043511  020110
 12991  105316  054502  042524  020054
 12992  105324  040504  040524  050040
 12993  105332  051101  052111  020131
 12994  105340  044103  041505  042513
 12995  105346  026122
 12996  105350  051200  040505  044504            .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
 12997  105356  043516  040440  042040
 12998  105364  052101  020101  040520
 12999  105372  052124  051105  020116
 13000  105400  044127  041511  020110
 13001  105406  044123  052517  042114
 13002  105414  044040  053101  020105
 13003  105422  040503  051525  042105            .ASCIZ  'CAUSED AN ERROR.'
 13004  105430  040440  020116  051105
 13005  105436  047522  027122     000
 13006
 13007  105443                          EM147:
 13008  105443     103  041501  042510            .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
 13009  105450  042040  052101  020101
 13010  105456  042515  047515  054522
 13011  105464  050040  051101  052111
 13012  105472  020131  044103  041505
 13013  105500  042513  051522  052040
 13014  105506  051505  020124  040506
 13015  105514  046111  042105     056
 13016  105521     200  047125  041101            .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
 13017  105526  042514  052040  020117
 13018  105534  047506  041522  020105
 13019  105542  020101  040520  044522
 13020  105550  054524  042440  051122
 13021  105556  051117  020054  051525
 13022  105564  047111  020107
 13023  105570  044124  020105  040515            .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
```

```
105576  047111  042524  040516
105604  041516  020105  042522
105612  044507  052123  051105
105620  100054
105622  052101  052040  042510          .ASCII  'AT THE GROUP ONE,HIGH BYTE, DATA PARITY CHECKER,'
105630  043440  047522  050125
105636  047440  042516  044054
105644  043511  020110  054502
105652  042524  020054  040504
105660  040524  050040  051101
105666  052111  020131  044103
105674  041505  042513  026122
105702  051200  040505  044504          .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
105710  043516  040440  042040
105716  052101  020101  040520
105724  052124  051105  020116
105732  044127  041511  020110
105740  044123  052517  042114
105746  044040  053101  020105
105754  040503  051525  042105          .ASCIZ  'CAUSED AN ERROR.'
105762  040440  020116  051105
105770  047522  027122     000

105775     200  047125  054105  EM150:  .ASCIZ  <CRLF>'UNEXPECTED CPU ERROR TRAPPED TO VECTOR ERRVEC (4)!'
106002  042520  052103  042105
106010  041440  052520  042440
106016  051122  051117  052040
106024  040522  050120  042105
106032  052040  020117  042526
106040  052103  051117  042440
106046  051122  042526  020103
106054  032050  020451     000

106061     115  051501  020123  EM151:  .ASCIZ  'MASS BUS WRITE HIT DID NOT INVALIDATE THE CACHE.'
106066  052502  020123  051127
106074  052111  020105  044510
106102  020124  044504  020104
106110  047516  020124  047111
106116  040526  044514  040504
106124  042524  052040  042510
106132  041440  041501  042510
106140  000056

        106061                  EM152=EM151
        106061                  EM153=EM151

106142  042504  044526  042503  EM154:  .ASCIZ  'DEVICE ERROR IN THE RS04.'
106150  042440  051122  051117
106156  044440  020116  044124
106164  020105  051522  032060
106172  000056

106174  042504  044526  042503  EM155:  .ASCIZ  'DEVICE ERROR IN THE RP04.'
106202  042440  051122  051117
106210  044440  020116  044124
106216  020105  050122  032060
```

```
 13080  106224  000056
 13081
 13082  106226  042504  044526  042503  EM156:  .ASCIZ  'DEVICE ERROR IN THE MASS BUS TESTER.'
 13083  106234  042440  051122  051117
 13084  106242  044440  020116  044124
 13085  106250  020105  040515  051523
 13086  106256  041040  051525  052040
 13087  106264  051505  042524  027122
 13088  106272     000
 13089
 13090
 13091  106273     104  053105  041511  EM160:  .ASCIZ  'DEVICE ERROR IN THE RK05.'
 13092  106300  020105  051105  047522
 13093  106306  020122  047111  052040
 13094  106314  042510  051040  030113
 13095  106322  027065     000
 13096
 13097  106325     104  053105  041511  EM161:  .ASCIZ  'DEVICE ERROR IN THE UNIBUS EXERCISER.'
 13098  106332  020105  051105  047522
 13099  106340  020122  047111  052040
 13100  106346  042510  052440  044516
 13101  106354  052502  020123  054105
 13102  106362  051105  044503  042523
 13103  106370  027122     000
 13104  106373     125  041502  020102  EM162:  .ASCII  /UBCB PE ABORT DOESN'T GO LOW OR/<CRLF>
 13105  106400  042520  040440  047502
 13106  106406  052122  042040  042517
 13107  106414  047123  052047  043440
 13108  106422  020117  047514  020127
 13109  106430  051117     200
 13110  106433     111  020124  047504           .ASCII  /IT DOESN'T GET TO TMCC E33 OR E33 BAD/<CRLF>
 13111  106440  051505  023516  020124
 13112  106446  042507  020124  047524
 13113  106454  052040  041515  020103
 13114  106462  031505  020063  051117
 13115  106470  042440  031463  041040
 13116  106476  042101     200
 13117  106501     117  020122  041125           .ASCII  /OR UBCB PARITY ERR DOESN'T GET TO TMCB E53/<CRLF>
 13118  106506  041103  050040  051101
 13119  106514  052111  020131  051105
 13120  106522  020122  047504  051505
 13121  106530  023516  020124  042507
 13122  106536  020124  047524  052040
 13123  106544  041515  020102  032505
 13124  106552  100063
 13125  106554  051501  040440  046040           .ASCIZ  /AS A LOW OR E53(5) BAD/
 13126  106562  053517  047440  020122
 13127  106570  032505  024063  024465
 13128  106576  041040  042101     000
 13129  106603     125  041502  020102  EM163:  .ASCII  /UBCB PARITY ERR DOESN'T GO LOW OR IT DOES/<CRLF>
 13130  106610  040520  044522  054524
 13131  106616  042440  051122  042040
 13132  106624  042517  047123  052047
 13133  106632  043440  020117  047514
 13134  106640  020127  051117  044440
 13135  106646  020124  047504  051505
 13136  106654     200
 13137  106655     116  052117  043440             .ASCIZ  /NOT GET TO DAPE/
 13138  106662  052105  052040  020117
 13139  106670  040504  042520     000
 13140  106675     104  050101  020105  EM164:  .ASCIZ  /DAPE E11(4) BAD OR TV06 DOESN'T GET TO THE ALU/
 13141  106702  030505  024061  024464
 13142  106710  041040  042101  047440
 13143  106716  020122  053124  033060
 13144  106724  042040  042517  047123
 13145  106732  052047  043440  052105
 13146  106740  052040  020117  044124
 13147  106746  020105  046101  000125
 13148  106754  040504  042520  042440  EM165:  .ASCIZ  /DAPE E7(1) BAD/
 13149  106762  024067  024461  041040
 13150  106770  042101     000
 13151  106773     124  041515  020101  EM166:  .ASCIZ  /TMCA SEG+CON+PAR DOESN'T GO LOW ON CCBJ PARITY TRAP/
 13152  107000  042523  025507  047503
 13153  107006  025516  040520  020122
 13154  107014  047504  051505  023516
 13155  107022  020124  047507  046040
 13156  107030  053517  047440  020116
 13157  107036  041503  045102  050040
 13158  107044  051101  052111  020131
 13159  107052  051124  050101     000
 13160  107057     124  041515  020102  EM167:  .ASCII  /TMCB PART DOESN'T GO LOW OR DOES/<CRLF>
 13161  107064  040520  052122  042040
 13162  107072  042517  047123  052047
 13163  107100  043440  020117  047514
 13164  107106  020127  051117  042040
 13165  107114  042517  100123
 13166  107120  047516  020124  042507           .ASCIZ  /NOT GET TO UBCB OR UBCB E18(1) BAD/
 13167  107126  020124  047524  052440
 13168  107134  041502  020102  051117
 13169  107142  052440  041502  020102
 13170  107150  030505  024070  024461
 13171  107156  041040  042101     000
 13172  107163     124  041515  020101  EM170:  .ASCIZ  /TMCA E67(8) DOESN'T GO LOW ON MGMT/
 13173  107170  033105  024067  024470
 13174  107176  042040  042517  047123
 13175  107204  052047  043440  020117
 13176  107212  047514  020127  047117
 13177  107220  046440  046507  000124
 13178  107226  046524  040503  042440  EM171:  .ASCIZ  /TMCA E67(12) DOESN'T GO LOW ON MGMT/
 13179  107234  033466  030450  024462
 13180  107242  042040  042517  047123
 13181  107250  052047  043440  020117
 13182  107256  047514  020127  047117
 13183  107264  046440  046507  000124
 13184  107272  046524  040503  042440  EM172:  .ASCII  /TMCA E68(6) DOESN'T GO LOW ON PAR TRP/<CRLF>
 13185  107300  034066  033050  020051
 13186  107306  047504  051505  023516
 13187  107314  020124  047507  046040
 13188  107322  053517  047440  020116
 13189  107330  040520  020122  051124
 13190  107336  100120
 13191  107340  051117  042440  032464           .ASCIZ  /OR E45(4) BAD/
```

```
107346  032050  020051  040502
107354  000104
107356  046524  040503  042440  EM173:  .ASCIZ  /TMCA E68(8) DOESN'T GO LOW ON PAR TRP/
107364  034066  034050  020051
107372  047504  051505  023516
107400  020124  047507  046040
107406  053517  047440  020116
107414  040520  020122  051124
107422  000120
107424  046524  041503  050040  EM435:  .ASCII  /TMCC PRIORITY CLEAR DIDN'T GO LOW OR DIDN'T/<CRLF>
107432  044522  051117  052111
107440  020131  046103  040505
107446  020122  044504  047104
107454  052047  043440  020117
107462  047514  020127  051117
107470  042040  042111  023516
107476  100124
107500  042507  020124  044124          .ASCIZ  /GET THRU TMCA E43(2) ON ABORT CLEAR/
107506  052522  052040  041515
107514  020101  032105  024063
107522  024462  047440  020116
107530  041101  051117  020124
107536  046103  040505  000122
107544  052502  020123  041120  EM174:  .ASCIZ  /BUS PB DIDN'T GET TO UBCB PE ABORT/
107552  042040  042111  023516
107560  020124  042507  020124
107566  047524  052440  041502
107574  020102  042520  040440
107602  047502  052122     000
107607     125  041502  020102  EM175:  .ASCIZ  /UBCB PARITY ERR DIDN'T GO LOW ON BUS PB/
107614  040520  044522  054524
107622  042440  051122  042040
107630  042111  023516  020124
107636  047507  046040  053517
107644  047440  020116  052502
107652  020123  041120     000

                                ;THESE ARE DATA HEADERS:

107657     040  052040  051505  DH1:    .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'PHYSICAL ADDR.'<TAB>'CALL AT PC.'
107664  027124  020011  051107
107672  052517  027120  050011
107700  054510  044523  040503
107706  020114  042101  051104
107714  004456  040503  046114
107722  040440  020124  041520
107730  000056

107732  020040  042524  052123  DH2:    .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR ADDR REG.'<TAB>'ERROR REG.'<TAB>
107740  004456  043440  047522
107746  050125  004456  051105
107754  047522  020122  042101
107762  051104  051040  043505
107770  004456  051105  047522
107776  020122  042522  027107
110004     011
```

```
110005     122  043105  040440                  .ASCIZ  'REF ADDR.'<TAB>'TRAP AT PC.'
110012  042104  027122  052011
110020  040522  020120  052101
110026  050040  027103     000

        107732                         DH3=DH2

        107732                         DH4=DH2

110033     040  052040  051505  DH5:    .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'READ.'
110040  027124  041411  046101
110046  020114  052101  050040
110054  027103  051011  040505
110062  027104     000

        110033                         DH6=DH5

        110033                         DH7=DH5

        110033                         DH10=DH5

110065     040  052040  051505  DH11:   .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'TRAP AT PC.'<TAB>'ERROR ADDR REG.'
110072  027124  020011  051107
110100  052517  027120  052011
110106  040522  020120  052101
110114  050040  027103  042411
110122  051122  051117  040440
110130  042104  020122  042522
110136  027107     000

110141     040  052040  051505  DH12:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'CALL AT PC.'<TAB>'TEST ADDR.'<TAB>
110146  027124  020011  051107
110154  052517  027120  041411
110162  046101  020114  052101
110170  050040  027103  052011
110176  051505  020124  042101
110204  051104  004456
110210  040504  040524  053440          .ASCIZ  'DATA WR. DATA READ.'
110216  027122  042040  052101
110224  020101  042522  042101
110232  000056

110234  020040  042524  052123  DH13:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'*DATA.'<TAB>'+DATA.'<TAB>
110242  004456  043440  047522
110250  050125  004456  042052
110256  052101  027101  025411
110264  040504  040524  004456
110272  051105  047522  020122          .ASCIZ  'ERROR COUNT.'
110300  047503  047125  027124
110306     000

110307     040  052040  051505  DH14:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADDR REG.'
110314  027124  041411  046101
110322  020114  052101  050040
110330  027103  042411  051122
110336  051117  040440  042104
```

```
13304  110344  020122  042522  027107
13305  110352  052011  040522  020120            .ASCII  <TAB>'TRAP AT PC.'<TAB>
13306  110360  052101  050040  027103
13307  110366     011
13308  110367     105  051122  051117            .ASCIZ  'ERROR REG.'
13309  110374  051040  043505  000056
13310
13311  110402  020040  042524  052123  DH15:     .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
13312  110410  004456  040503  046114
13313  110416  040440  020124  041520
13314  110424  000056
13315
13316  110426  020040  042524  052123  DH16:     .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'WROTE.'<TAB>'READ.'<TAB>
13317  110434  004456  043440  047522
13318  110442  050125  004456  051127
13319  110450  052117  027105  051011
13320  110456  040505  027104     011
13321  110463     101  042104  020122            .ASCIZ  'ADDR TESTED.'<TAB>'CALL AT PC.'
13322  110470  042524  052123  042105
13323  110476  004456  040503  046114
13324  110504  040440  020124  041520
13325  110512  000056
13326
13327  110514  020040  042524  052123  DH17:     .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR AT PC.'<TAB>'READ.'<TAB>
13328  110522  004456  043440  047522
13329  110530  050125  004456  051105
13330  110536  047522  020122  052101
13331  110544  050040  027103  051011
13332  110552  040505  027104     011
13333  110557     111  027116  040411            .ASCIZ  'IN.'<TAB>'ADDRESS.'
13334  110564  042104  042522  051523
13335  110572  000056
13336
13337          110514                  DH20=DH17
13338
13339  110574  020040  042524  052123  DH21:     .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'READ.'<TAB>' GROUP.'<TAB>'ADDRESS.'
13340  110602  004456  040503  046114
13341  110610  040440  020124  041520
13342  110616  004456  042522  042101
13343  110624  004456  043440  047522
13344  110632  050125  004456  042101
13345  110640  051104  051505  027123
13346  110646     000
13347
13348  110647     040  052040  051505  DH22:     .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'EXPECTED ERROR AT.'
13349  110654  027124  041411  046101
13350  110662  020114  052101  050040
13351  110670  027103  042411  050130
13352  110676  041505  042524  020104
13353  110704  051105  047522  020122
13354  110712  052101  000056
13355
13356  110716  020040  042524  052123  DH23:     .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'EXPECTED ADRS.'<TAB>
13357  110724  004456  040503  046114
13358  110732  040440  020124  041520
13359  110740  004456  054105  042520
```

```
110746  052103  042105  040440
110754  051104  027123     011
110761     107  052117  040440          .ASCIZ  'GOT ADRS.'<TAB>'ERROR REG.'
110766  051104  027123  042411
110774  051122  051117  051040
111002  043505  000056

        110647                  DH24=DH22

        110716                  DH25=DH23

111006  020040  042524  052123  DH26:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
111014  004456  040503  046114
111022  040440  020124  041520
111030  004456  043440  047522
111036  050125  004456  042101
111044  051104  051505  027123
111052     000

111053     040  052040  051505  DH27:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ESTABLISHED HIT.'
111060  027124  041411  046101
111066  020114  052101  050040
111074  027103  020011  051107
111102  052517  027120  042411
111110  052123  041101  044514
111116  044123  042105  044040
111124  052111     056
111127     040  052502  020124          .ASCIZ  ' BUT GOT HIT.'
111134  047507  020124  044510
111142  027124     000

        110647                  DH30=DH22

        110716                  DH31=DH23

        110647                  DH32=DH22

        110716                  DH33=DH23

111145     040  052040  051505  DH34:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'READ.'<TAB>'IN ADDRESS.'<TAB>
111152  027124  050011  020103
111160  043117  041440  046101
111166  027114  051011  040505
111174  027104  044411  020116
111202  042101  051104  051505
111210  027123     011
111213     105  050130  041505          .ASCIZ  'EXPECTED.'
111220  042524  027104     000

        111145                  DH35=DH34

111225     040  052040  051505  DH36:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
111232  027124  041411  046101
111240  020114  052101  050040
111246  027103  020011  051107
111254  052517  027120  040411
```

```
13416  111262  042104  042522  051523
13417  111270  000056
13418
13419  111272  020040  042524  052123  DH37:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR COUNT.'<TAB>
13420  111300  004456  043440  047522
13421  111306  050125  004456  051105
13422  111314  047522  020122  047503
13423  111322  047125  027124     011
13424  111327     052  041040  042101          .ASCIZ  '* BAD ADRS.'<TAB>'+ BAD ADRS.'
13425  111334  040440  051104  027123
13426  111342  025411  041040  042101
13427  111350  040440  051104  027123
13428  111356     000
13429
13430
13431  111357     040  052040  051505  DH41:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
13432  111364  027124  041411  046101
13433  111372  020114  052101  050040
13434  111400  027103  020011  051107
13435  111406  052517  027120  040411
13436  111414  042104  042522  051523
13437  111422  000056
13438
13439          111357                  DH42=DH41
13440
13441  111424  020040  042524  052123  DH43:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>' GROUP.'
13442  111432  004456  040503  046114
13443  111440  040440  020124  041520
13444  111446  004456  051124  050101
13445  111454  040440  020124  041520
13446  111462  004456  043440  047522
13447  111470  050125     056
13448
13449          111424                  DH40=DH43
13450
13451  111473     040  052040  051505  DH44:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
13452  111500  027124  041411  046101
13453  111506  020114  052101  050040
13454  111514  027103  052011  040522
13455  111522  020120  052101  050040
13456  111530  027103     011
13457  111533     105  051122  051117          .ASCIZ  'ERROR ADRS REG.'<TAB>'ERROR REG.'
13458  111540  040440  051104  020123
13459  111546  042522  027107  042411
13460  111554  051122  051117  051040
13461  111562  043505  000056
13462
13463          111473                  DH45=DH44
13464
13465  111566  020040  042524  052123  DH46:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
13466  111574  004456  040503  046114
13467  111602  040440  020124  041520
13468  111610  000056
13469
13470          111566                  DH47=DH46
13471
```

```
        111473                  DH50=DH44

        111473                  DH51=DH44

        111566                  DH52=DH46

        111566                  DH53=DH46

111612  020040  042524  052123  DH54:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR COUNT.'
111620  004456  040503  046114
111626  040440  020124  041520
111634  004456  051105  047522
111642  020122  047503  047125
111650  027124     000
111653     040  050040  020103  DH55:   .ASCIZ  /  PC     CCR/
111660  020040  041440  051103
111666     000
111667     040  050040  020103  DH66:   .ASCIZ  /  PC     CCR   GROUP   TST-DATA-ADRS/
111674  020040  041440  051103
111702  020040  043440  047522
111710  050125  020040  052040
111716  052123  042055  052101
111724  026501  042101  051522
111732     000
111733     040  050040  020103  DH71:   .ASCIZ  /  PC   EXPCTD  RECVD  LOC/
111740  020040  054105  041520
111746  042124  020040  042522
111754  053103  020104  046040
111762  041517     000
111765     040  050040  020103  DH107:  .ASCIZ  /  PC     CCR    MSER   GROUP   EXPCTD-B4,5/
111772  020040  041440  051103
112000  020040  020040  051515
112006  051105  020040  043440
112014  047522  050125  020040
112022  042440  050130  052103
112030  026504  032102  032454
112036     000
112037     040  050040  020103  DH111:  .ASCIZ  /  PC     MSER/
112044  020040  046440  042523
112052  000122
112054  020040  041520  020040  DH115:  .ASCIZ  /  PC     HITMIS/
112062  020040  044510  046524
112070  051511     000

112073     040  050040  020103  DH123:  .ASCIZ  ?  PC     KIPDR  (KIPDR)?
112100  020040  045440  050111
112106  051104  020040  045450
112114  050111  051104  000051
112122  020040  041520  020040  DH125:  .ASCIZ  /  PC     CCR  PAR-ADR   (PAR)  (PDR)  TST-DATA-ADRS(VA)/
112130  020040  041503  020122
112136  050040  051101  040455
112144  051104  020040  024040
112152  040520  024522  020040
112160  050050  051104  020051
112166  052040  052123  042055
112174  052101  026501  042101
```

```
112202  051522  053050  024501
112210     000
112211     040  050040  020103  DH130:  .ASCIZ  ?  PC     SIPDR  (SIPDR)?
112216  020040  051440  050111
112224  051104  020040  051450
112232  050111  051104  000051
112240  020040  041520  020040  DH132:  .ASCIZ  ?  PC     UIPDR  (UIPDR)?
112246  020040  044525  042120
112254  020122  024040  044525
112262  042120  024522     000


112267     040  052040  051505  DH136:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
112274  027124  041411  046101
112302  020114  052101  050040
112310  027103  020011  051107
112316  052517  027120  040411
112324  042104  042522  051523
112332  000056

        112267                  DH137=DH136

112334  020040  042524  052123  DH140:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'DATA.'<TAB>'ADDRESS.'
112342  004456  040503  046114
112350  040440  020124  041520
112356  004456  040504  040524
112364  004456  042101  051104
112372  051505  027123     000

        112334                  DH141=DH140

        112334                  DH142=DH140

        112334                  DH143=DH140

        112334                  DH144=DH140

        112334                  DH145=DH140

        112334                  DH146=DH140

        112334                  DH147=DH140

112377     040  052040  051505  DH150:  .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'CPU ERROR REGISTER.'
112404  027124  052011  040522
112412  020120  052101  050040
112420  027103  041411  046101
112426  020114  052101  050040
112434  027103  041411  052520
112442  042440  051122  051117
112450  051040  043505  051511
112456  042524  027122     000

112463     125  044523  043516  DH151:  .ASCII  'USING THE RS04.'
112470  052040  042510  051040
112476  030123  027064
```

```
112502  020040  042524  052123          .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
112510  004456  051107  052517
112516  027120  040411  042104
112524  042522  051523  000056

112532  051525  047111  020107  DH152:  .ASCII  'USING THE RP04.'
112540  044124  020105  050122
112546  032060     056
112551     040  052040  051505          .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
112556  027124  043411  047522
112564  050125  004456  042101
112572  051104  051505  027123
112600     000

112601     125  044523  043516  DH153:  .ASCII  'USING THE MASS BUS TESTER.'
112606  052040  042510  046440
112614  051501  020123  052502
112622  020123  042524  052123
112630  051105     056
112633     040  052040  051505          .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
112640  027124  043411  047522
112646  050125  004456  042101
112654  051104  051505  027123
112662     000

112663     040  052040  051505  DH154:  .ASCIZ  '  TEST.'<TAB>'RS4CS2.'<TAB>'RS4DS.'<TAB>'RS4ER.'
112670  027124  051011  032123
112676  051503  027062  051011
112704  032123  051504  004456
112712  051522  042464  027122
112720     000

112721     040  052040  051505  DH155:  .ASCIZ  '  TEST.'<TAB>'RP4CS2.'<TAB>'RP4DS.'<TAB>'RP4ER.'
112726  027124  051011  032120
112734  051503  027062  051011
112742  032120  051504  004456
112750  050122  042464  027122
112756     000

112757     040  052040  051505  DH156:  .ASCIZ  '  TEST.'<TAB>'RH4CS2.'<TAB>'RH4ST.'<TAB>'RH4ER.'
112764  027124  051011  032110
112772  051503  027062  051011
113000  032110  052123  004456
113006  044122  042464  027122
113014     000


113015     040  052040  051505  DH160:  .ASCIZ  '  TEST.'<TAB>'RK5ER.'<TAB>'RK5DS.'
113022  027124  051011  032513
113030  051105  004456  045522
113036  042065  027123     000

113043     040  052040  051505  DH161:  .ASCIZ  '  TEST.'<TAB>'UBECR1.'<TAB>'UBECR2.'
113050  027124  052411  042502
113056  051103  027061  052411
113064  042502  051103  027062
```

```
113072     000
113073     105  051122  051117  DH162:  .ASCIZ  /ERRORPC TEST NUMBER/
113100  041520  052040  051505
113106  020124  052516  041115
113114  051105     000

                                ;THESE ARE DATA FORMAT DESIGNATORS FOR THE DATA TABLE:

113117     004     004     003  DF1:    .BYTE   4,4,3,3
113122     003

113123     004     004     007  DF2:    .BYTE   4,4,7,0,3,3
113126     000     003     003

        113123                  DF3=DF2

        113123                  DF4=DF2

113131     004     003     000  DF5:    .BYTE   4,3,0,5,0,0,0,0
113134     005     000     000
113137     000     000

        113131                  DF6=DF5

        113131                  DF7=DF5

        113131                  DF10=DF5

113141     004     004     003  DF11:   .BYTE   4,4,3,7,5,0,5,3,0
113144     007     005     000
113147     005     003     000

113152     004     004     003  DF12:   .BYTE   4,4,3,3,0,0
113155     003     000     000

113160     004     004     000  DF13:   .BYTE   4,4,0,0,4
113163     000     004

113165     004     003     007  DF14:   .BYTE   4,3,7,3,0
113170     003     000

113172     004     003          DF15:   .BYTE   4,3

113174     004     004     000  DF16:   .BYTE   4,4,0,0,3,3
113177     000     003     003

113202     004     004     003  DF17:   .BYTE   4,4,3,0,5,3,5,5,5,3,5,3,5,3,5,3,5,0,5,0,5,0,5,0
113205     000     005     003
113210     005     005     005
113213     003     005     003
113216     005     003     005
113221     003     005     000
113224     005     000     005
113227     000     005     000

        113202                  DF20=DF17
```

```
13696
13697  113232       004      003      000  DF21:    .BYTE    4,3,0,4,3,5
13698  113235       004      003      005
13699  113240       005      003      005           .BYTE    5,3,5,3,5,3;5,3,5
13700  113243       003      005      003
13701  113246       005      003      005
13702  113251       000      005      000           .BYTE    0,5,0,5,0,5,0
13703  113254       005      000      005
13704  113257       000
13705
13706  113260       004      003      002  DF22:    .BYTE    4,3,2
13707
13708  113263       004      003      002  DF23:    .BYTE    4,3,2,2,0
13709  113266       002      000
13710
13711            113260                     DF24=DF22
13712
13713            113263                     DF25=DF23
13714
13715  113270       004      003      004  DF26:    .BYTE    4,3,4,2
13716  113273       002
13717
13718  113274       004      003      004  DF27:    .BYTE    4,3,4,2,2
13719  113277       002      002
13720
13721            113260                     DF30=DF22
13722
13723            113263                     DF31=DF23
13724
13725            113260                     DF32=DF22
13726
13727            113263                     DF33=DF23
13728
13729  113301       004      003      000  DF34:    .BYTE    4,3,0,2,0
13730  113304       002      000
13731
13732            113301                     DF35=DF34
13733
13734  113306       004      003      004  DF36:    .BYTE    4,3,4,2
13735  113311       002
13736
13737  113312       004      004      007  DF37:    .BYTE    4,4,7,2,2,0
13738  113315       002      002      000
13739
13740
13741  113320       004      003      004  DF41:    .BYTE    4,3,4,2
13742  113323       002
13743
13744            113320                     DF42=DF41
13745
13746  113324       004      003      003  DF43:    .BYTE    4,3,3,4,5,2,7,0
13747  113327       004      005      002
13748  113332       007      000
13749
13750            113324                     DF40=DF43
13751
```

```
 13752  113334     004      003      002  DF44:    .BYTE    4,3,2,7,0,5,2,5,0,5,2,5,0,5,2
 13753  113337     007      000      005
 13754  113342     002      005      000
 13755  113345     005      002      005
 13756  113350     000      005      002
 13757
 13758          113334                     DF45=DF44
 13759
 13760  113353     004      003      005  DF46:    .BYTE    4,3,5,2,5,0,5,2,5,0,5,2
 13761  113356     002      005      000
 13762  113361     005      002      005
 13763  113364     000      005      002
 13764
 13765          113353                     DF47=DF46
 13766
 13767          113334                     DF50=DF44
 13768
 13769          113334                     DF51=DF44
 13770
 13771          113353                     DF52=DF46
 13772
 13773          113353                     DF53=DF46
 13774
 13775  113367     004      003      004  DF54:    .BYTE    4,3,4
 13776  113372     000      000           DF55:    .BYTE    0,0
 13777          113372                     DF56=DF55
 13778          113372                     DF57=DF55
 13779          113372                     DF60=DF55
 13780          113372                     DF61=DF55
 13781          113372                     DF62=DF55
 13782          113372                     DF63=DF55
 13783          113372                     DF64=DF55
 13784          113372                     DF65=DF55
 13785  113374     000      000      000  DF66:    .BYTE    0,0,0,0
 13786  113377     000
 13787          113374                     DF67=DF66
 13788          113374                     DF70=DF66
 13789          113374                     DF71=DF66
 13790          113374                     DF72=DF66
 13791          113374                     DF73=DF66
 13792          113374                     DF74=DF66
 13793          113374                     DF75=DF66
 13794          113374                     DF76=DF66
 13795          113374                     DF77=DF66
 13796
 13797  113400     000      000      000  DF100:   .BYTE    0,0,0,0,0,0
 13798  113403     000      000      000
 13799          113372                     DF103=DF55
 13800          113372                     DF104=DF55
 13801          113372                     DF105=DF55
 13802          113372                     DF106=DF55
 13803  113406     000      000      000  DF107:   .BYTE    0,0,0,0,0
 13804  113411     000      000
 13805
 13806          113372                     DF110=DF55
 13807          113372                     DF111=DF55
```

```
        113372                  DF112=DF55
        113374                  DF113=DF66
        113372                  DF115=DF55
113413     000     000     000  DF123:  .BYTE   0,0,0
113416     004     003     004  DF136:  .BYTE   4,3,4,2
113421     002
        113416                  DF137=DF136

113422     004     003     000  DF140:  .BYTE   4,3,0,2
113425     002

        113422                  DF141=DF140

        113422                  DF142=DF140

        113422                  DF143=DF140

        113422                  DF144=DF140

        113422                  DF145=DF140

        113422                  DF146=DF140

        113422                  DF147=DF140

113426     004     003     003  DF150:  .BYTE   4,3,3,0
113431     000

113432     004     004     007  DF151:  .BYTE   4,4,7

        113432                  DF152=DF151
        113432                  DF153=DF151

113435     004     000     000  DF154:  .BYTE   4,0,0,0
113440     000

        113435                  DF155=DF154
        113435                  DF156=DF154
        113435                  DF157=DF154
        113435                  DF160=DF154
        113435                  DF161=DF154


        113442                          .EVEN

                                ;THESE ARE DATA TABLES:

113442  001626  001630  001632  DT1:    .WORD   $TMP0,$TMP1,$TMP2,$ERRPC,0
113450  001520  000000

113454  001626  001642  001632  DT2:    .WORD   $TMP0,$TMP6,$TMP2,$TMP1,$TMP5,$TMP4,0
113462  001630  001640  001636
113470  000000

        113454                  DT3=DT2

        113454                    DT4=DT2

113472  001626  001520  001632    DT5:     .WORD    $TMP0,$ERRPC,$TMP2,MTA5,JJPAT1,JJPAT2,JJPAT3,JJPAT4,0
113500  065100  026132  026134
113506  026136  026140  000000

        113472                    DT6=DT5

        113472                    DT7=DT5

        113472                    DT10=DT5

113514  001626  001630  001632    DT11:    .WORD    $TMP0,$TMP1,$TMP2,$TMP4,MTA11,$TMP3,$TAB,$TMP7,$TMP6,0
113522  001636  065162  001634
113530  065076  001644  001642
113536  000000

113540  001626  001630  001520    DT12:    .WORD    $TMP0,$TMP1,$ERRPC,$TMP3,$TMP4,$TMP5,0
113546  001634  001636  001640
113554  000000

113556  001626  001630  001632    DT13:    .WORD    $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,0
113564  001634  001636  000000

113572  001626  001520  001630    DT14:    .WORD    $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,0
113600  001634  001636  000000

113606  001626  001630  000000    DT15:    .WORD    $TMP0,$TMP1,0

113614  001626  001630  001632    DT16:    .WORD    $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,$ERRPC,0
113622  001634  001636  001520
113630  000000

113632  001626  001630  001632    DT17:    .WORD    $TMP0,$TMP1,$TMP2,$TMP3,MTC17,$TMP4,$CRLF,MTB17
113640  001634  065254  001636
113646  001707  065234
113652  065227  001640  065227             .WORD    MTA17,$TMP5,MTA17,$TMP6,MTA17,$TMP7,MTA17,$TMP10
113660  001642  065227  001644
113666  065227  001646
113672  001707  035376  065076             .WORD    $CRLF,MMPAT1,$TAB,MMPAT2,$TAB,MMPAT3,$TAB,MMPAT4,0
113700  035400  065076  035402
113706  065076  035404  000000

113714  001626  001630  001632    DT20:    .WORD    $TMP0,$TMP1,$TMP2,$TMP3,MTA20,$TMP4,$CRLF,MTB17
113722  001634  065263  001636
113730  001707  065234
113734  001640  065227  001642             .WORD    $TMP5,MTA17,$TMP6,MTA17,$TMP7,MTA17,$TMP10,MTA17
113742  065227  001644  065227
113750  001646  065227
113754  001707  035376  065076             .WORD    $CRLF,MMPAT1,$TAB,MMPAT3,$TAB,MMPAT3,$TAB,MMPAT4,0
113762  035402  065076  035402
113770  065076  035404  000000

113776  001626  001630  001632    DT21:    .WORD    $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,MTA21
114004  001634  001636  065272
114012  065227  001640  065227             .WORD    MTB21,$TMP5,MTB21,$TMP6,MTB21,$TMP7,MTB21,$TMP10,$CRLF
```

```
114020  001642  065227  001644
114026  065227  001646  001707
114034  033532  065076  033534          .WORD   KKPAT1,$TAB,KKPAT2,$TAB,KKPAT3,$TAB,KKPAT4,0
114042  065076  033536  065076
114050  033540  000000

114054  001626  001520  007466  DT22:   .WORD   $TMP0,$ERRPC,XADR2,0
114062  000000

114064  001626  001520  007466  DT23:   .WORD   $TMP0,$ERRPC,XADR2,$TMP3,$TMP1,0
114072  001634  001630  000000

114100  001626  001520  010366  DT24:   .WORD   $TMP0,$ERRPC,XXADR2,0
114106  000000

114110  001626  001520  010366  DT25:   .WORD   $TMP0,$ERRPC,XXADR2,$TMP3,$TMP1,0
114116  001634  001630  000000

114124  001626  001520  001630  DT26:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP2,0
114132  001632  000000

114136  001626  001520  001630  DT27:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP2,$TMP4,0
114144  001632  001636  000000

114152  001626  001520  011304  DT30:   .WORD   $TMP0,$ERRPC,RRADR2,0
114160  000000

114162  001626  001520  011304  DT31:   .WORD   $TMP0,$ERRPC,RRADR2,$TMP3,$TMP1,0
114170  001634  001630  000000

114176  001626  001520  012166  DT32:   .WORD   $TMP0,$ERRPC,SSADR2,0
114204  000000

114206  001626  001520  012166  DT33:   .WORD   $TMP0,$ERRPC,SSADR2,$TMP3,$TMP1,0
114214  001634  001630  000000

114222  001626  001520  001632  DT34:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,$TMP5,0
114230  001634  001640  000000

        114222                  DT35=DT34

114236  001626  001520  001632  DT36:   .WORD   $TMP0,$ERRPC,$TMP2,BBADR1,0
114244  014772  000000

114250  001626  001630  015012  DT37:   .WORD   $TMP0,$TMP1,BBCNT1,BBADR2,BBADR3,0
114256  014776  015002  000000


114264  001626  001520  001632  DT41:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
114272  001634  000000

        114264                  DT42=DT41

114276  001626  001520  001632  DT43:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA43,$TMP5,$TMP7,$TMP4,0
114304  001634  065357  001640
114312  001644  001636  000000
```

```
        114276                  DT40=DT43

114320  001626  001520  001662  DT44:   .WORD   $TMP0,$ERRPC,$TMP16,$TMP3,$TMP5,MTA45,$TMP12,MTB45
114326  001634  001640  065432
114334  001652  065460
114340  001646  065475  001642          .WORD   $TMP10,MTC45,$TMP6,MTB45,$TMP11,MTC45,$TMP14,0
114346  065460  001650  065475
114354  001656  000000

        114320                  DT45=DT44

114360  001626  001652  065432  DT46:   .WORD   $TMP0,$TMP12,MTA45,$TMP10,MTB45,$TMP6,MTC45
114366  001646  065460  001642
114374  065475
114376  001632  065460  001644          .WORD   $TMP2,MTB45,$TMP7,MTC45,$TMP4,0
114404  065475  001636  000000

        114360                  DT47=DT46

114412  001626  001520  001662  DT50:   .WORD   $TMP0,$ERRPC,$TMP16,$TMP3,$TMP5,MTA50,$TMP12,MTB45
114420  001634  001640  065510
114426  001652  065460
114432  001646  065475  001642          .WORD   $TMP10,MTC45,$TMP6,MTB45,$TMP11,MTC45,$TMP14,0
114440  065460  001650  065475
114446  001656  000000

        114412                  DT51=DT50

114452  001626  001652  065510  DT52:   .WORD   $TMP0,$TMP12,MTA50,$TMP10,MTB45,$TMP6,MTC45
114460  001646  065460  001642
114466  065475
114470  001632  065460  001644          .WORD   $TMP2,MTB45,$TMP7,MTC45,$TMP4,0
114476  065475  001636  000000

        114452                  DT53=DT52

114504  001626  001520  001632  DT54:   .WORD   $TMP0,$ERRPC,$TMP2,0
114512  000000
114514  001520  001556  000000  DT55:   .WORD   $ERRPC,$REG0,0
114522  001520  001556  001560  DT66:   .WORD   $ERRPC,$REG0,$REG1,$REG2,0
114530  001562  000000
114534  001520  001556  001560  DT100:  .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,0
114542  001562  001564  001566
114550  000000
114552  001520  001556  001560  DT107:  .WORD   $ERRPC, $REG0,  $REG1, $REG2, $REG3,0
114560  001562  001564  000000

114566  001520  001556  001560  DT123:  .WORD   $ERRPC,$REG0,$REG1,0
114574  000000
114576  001520  001556  001560  DT125:  .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,0
114604  001562  001564  001566
114612  000000

114614  001626  001520  001632  DT136:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
114622  001634  000000
```

```

        114614                  DT137=DT136

114626  001626  001520  001632  DT140:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
114634  001634  000000

        114626                  DT141=DT140

        114626                  DT142=DT140

        114626                  DT143=DT140

        114626                  DT144=DT140

        114626                  DT145=DT140

        114626                  DT146=DT140

        114626                  DT147=DT140

114640  001626  001630  001632  DT150:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
114646  001634  000000

114652  001626  001630  001632  DT151:  .WORD   $TMP0,$TMP1,$TMP2,0
114660  000000

        114652                  DT152=DT151
        114652                  DT153=DT151
114662  001626  001630  001632  DT154:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
114670  001634  000000
        114662                  DT155=DT154
        114662                  DT156=DT154
        114652                  DT157=DT151
        114652                  DT160=DT151
        114652                  DT161=DT151
114674  001520  001502  000000  DT162:  .WORD   $ERRPC,$TSTNM,0

        114702                          TLOC=.
        114700                          TLOC=-4&TLOC
        114704                          TLOC=TLOC+4
        114704                          .=TLOC
114704  001000                  TSTDAT: .BLKW   512.



116704  000000  000000  000000  BOTTOM: .WORD   0,0,0
        124704                          BOTPRG=BOTTOM+6000
        000001                          .END
```

```
AA     = 000015              3323#
AAADR1   016214              3364*   3365*   3374    3375    3486*   3487*   3508#
AAADR2   016224              3374*   3375*   3376*   3377*   3378*   3388    3390    3411    3412    3464    3465    3482    3483
                             3512#   3533    3535
AADONE   016372              3491    3562#
AAERGS   016212              3361*   3451    3476    3495*   3507#
AAERR1   016240              3447    3519#
AAERR2   016330              3541    3550#
AAEXER   016230              3362*   3492*   3514#   3523
AAFLG1   016204              3360*   3439    3463    3481    3490    3494*   3498#
AAFLG2   016206              3405*   3474*   3501#   3554    3557
AAGS     016210              3359*   3368    3493*   3505#
AAHIAD   015426              3348#
AALOAD   015424              3347#   3349*   3388    3390
AAOFST   016220              3353*   3354*   3376    3378    3510#
AATMP1   016232              3515#
AATMP2   016234              3351    3516#
AA1      015520              3364#   3496
AA16     016100              3480#
AA2      015540              3371#   3488
AA3      015632              3396    3401    3405#
AA4      016002              3436    3445#
AA5      016030              3462#
AA6      016054              3468#   3559
AA7      016124              3443    3486#   3556
AA8      016142              3399    3490#
ABORTT   054202              9704    9928#
ADRNG    066013              9974   11473#
ASLOC    047076              8710*   8716    8725    8731    8738#
ASRBCB   046732              8701#
AVMBL    042056              7628*   7634#   7654
BACKAD   054320              9944*   9953    9954#
BB     = 000014              3083#
BBADR1   014772              3113    3149    3151    3166    3187*   3188*   3204#   3225    3227    3253    3254    3257    3258
                             3260    3263   13961
BBADR2   014776              3206#   3257*   3258*  13964
BBADR3   015002              3208#   3259*   3260*   3261*   3262*   3263*   3264*  13964
BBCNT1   015012              3141*   3142*   3216#   3267*   3268*  13964
BBDONE   015302              3198    3282#
BBERR1   015022              3144    3222#   3279
BBERR2   015060              3233    3242#
BBERR3   015074              3243    3246#
BBERR4   015260              3270    3277#
BBERR5   015262              3276    3278#
BBFLG1   015006              3096*   3184    3194    3197    3199*   3211#   3247
BBFLG2   015010              3140*   3191    3213#   3266*
BBGM     015020              3098*   3105    3201*   3220#
BBGS     015016              3097*   3103    3109    3200*   3219#
BBHIAD   014306              3094#
BBLOAD   014304              3093#   3149    3151
BB0      014264              3084#   3101
BB1      014330              3100#   3202
BB2      014346              3103#   3107
BB3      014406              3113#
BB4      014554              3146#   3189
BB5      014612              3157    3162    3166#
```

```
BB6      014674          3182     3187#    3280
BB7      014710          3160     3191#
BB8      014736          3192     3197#
BIT0  = 000001           139#     438     1351     1352     1359     1675     5327     5481     9843
BIT00 = 000001           129#     139
BIT01 = 000002           128#     138
BIT02 = 000004           127#     137
BIT03 = 000010           126#     136
BIT04 = 000020           125#     135
BIT05 = 000040           124#     134
BIT06 = 000100           123#     133
BIT07 = 000200           122#     132     1301
BIT08 = 000400           121#     131     9275
BIT09 = 001000           120#     130     9286     9352
BIT1  = 000002           138#     437
BIT10 = 002000           119#     430     9335
BIT11 = 004000           118#     429     1354     1355     1357     1559     9293    10705    10866
BIT12 = 010000           117#     428    10701    10712    10864    11327
BIT13 = 020000           116#     427     9342
BIT14 = 040000           115#     426     1346     1347     1349     1538     8273    10703    10714    10871    11329    11335
BIT15 = 100000           114#     425      440     1360     1361     1363     1365     1366     1368     8253     8276     8280     9846
BIT2  = 000004           137#     436      442      443      444
BIT3  = 000010           136#     435      442      443      444
BIT4  = 000020           135#     434      443     8810     8814
BIT5  = 000040           134#     433      442     8812     8814    10681    10847    11050    11309
BIT6  = 000100           133#    1430    10072    10632    10699    10791    10958    11056    11132    11264
BIT7  = 000200           132#   10049    10498    10499    10506
BIT8  = 000400           131#     432     1380    10697    10862    11325
BIT9  = 001000           130#     431     9277    10707
BOTPRG= 124704          2188     2583     2619     2624     2639     2708     2723     2747     3114    14078#
BOTTOM  116704          1421    10055    14077#   14078
BPTVEC= 000014           146#
BYP   = 100000           440#    8949     8955     8956     8962     8963     8985     8991     8992     8998     8999     9020     9026
                        9027     9033     9034     9135     9166
CACHVE= 000114           153#    1444*    1495*    1575*    1576*    1601*    1602*    1626*    1673*    1769*    1896*    1997*    2129*
                        2213*    2323*    2416*    2523*    2604*    2676*    2807*    3100*    3144*    3270*    3279*    3328*    3422*
                        3447*    3525*    3595*    3646*    3825*    3835*    3891*    3944*    4123*    4133*    4175*    4308*    4447*
                        4606*    4778*    4784*    4842*    5026*    5142*    5149*    5269*    5277*    5321*    5331*    5374*    5475*
                        5485*    5528*    5618*    5704*    5874*    6213*    9906*    9913*    9942*
CALRH4= 104446          7194     7214     7234     7252     7739     9716#
CALRK5= 104450          7363     7383     7403     7421     9717#   10476
CALRP4= 104444          7030     7050     7070     7088     7785     9715#   10449
CALRS4= 104442          6862     6882     6902     6920     7762     9714#   10421
CALUBE= 104452          7509     7525     9718#
CBP     046366          8591#    8600
CBPA    046400          8594#    8675     8681
CC    = 000020          4170#
CCDONE  022226          4228     4272     4290#
CCERR1  021734          4175     4235#
CCERR2  021762          4239     4241#
CCERR3  022146          4188     4199     4211     4223     4275#
CCTMP1  021726          4230#
CCTMP2  021730          4179     4231#
CC1     021516          4182#
CC10    021660          4212     4216#    4271
CC11    021710          4223#    4226
```

```
CC12    021716       4222     4225#
CC13    021722       4224     4228#
CC2     021540       4188#    4191
CC3     021546       4187     4190#
CC4     021554       4189     4192#    4263
CC5     021602       4199#    4202
CC6     021610       4198     4201#
CC7     021616       4200     4204#    4267
CC8     021646       4211#    4214
CC9     021654       4210     4213#
CHAINQ  054544       9231    10054#
CISP    001715        592#
CLEAN   054232       9703     9940#
CNRNG   066223       9988    11500#
CONCMS  064766      10053    11353#
CONFLG  054446       9984    10010#
CONFL2  054462      10016#
CONTRL= 177746        163#     446#    1343     1373*    1374     1442*    1800*    2027*    2242*    2439*    2606*    2678*    2882*
                     2884*    3103*    3105*    3111*    3368*    3612*    3614*    3620*    3713*    3824     3828*    3834*    3910*
                     3912*    3918*    4013*    4122     4126*    4132*    4177*    4310*    4471*    4631*    4869*    4874*    4880*
                     4947*    4960*    5036*    5038*    5043*    5060*    5153*    5155*    5165*    5178*    5330*    5376*    5484*
                     5530*    5645*    5647*    5655*    5668*    5876*    5911*    5914*    5917*    5983*    5985*    5987*    5989*
                     5991*    5993*    5995*    5997*    6002*    6004*    6006*    6008*    6010*    6012*    6014*    6016*    6020*
                     6022*    6039*    6041*    6059*    6061*    6078*    6080*    6104*    6185*    6189*    6239*    6242*    6245*
                     7828*    7830*    7836*    7843*    7927*    7928     7930     7933*    7934     7936     7939*    7941     7944*
                     7945     7947     7951     7954     7957*    7958     7960     7962     7973*    7974     7976     7979     7981*
                     7982     7984     7987*    7988     7990     8002*    8005     8007     8010     8013     8028*    8029     8031*
                     8032     8034     8037     8039*    8040     8042     8045     8057*    8058     8060*    8061*    8062     8064
                     8068     8070*    8071     8073*    8074     8076     8080     8105*    8106     8114*    8116*    8121*    8122*
                     8125*    8126*    8127*    8128     8132     8135*    8140*    8146     8161     8164*    8165     8167*    8172*
                     8182     8222*    8223     8229*    8231*    8239*    8244*    8251     8308*    8309     8317*    8319*    8324*
                     8325*    8328*    8329*    8330*    8331     8335     8338*    8343*    8349     8364     8367*    8368     8370*
                     8375*    8385     8424*    8425     8430*    8432*    8442*    8446*    8447*    8452*    8460     8482*    8485*
                     8495*    8500*    8506     8546*    8547     8551*    8552*    8596*    8597     8602*    8604*    8613*    8616*
                     8618*    8626     8645*    8652     8703*    8705*    8708*    8714     8729     8756*    8757     8760*    8762*
                     8767*    8769*    8773*    8779*    8780     8793     8795     8800     8802     8817     8828     8832     8880*
                     8881     8884*    8886*    8888*    8889*    8897     8904     8907     8913*    8914     8916     8933*    9064*
                     9065     9070*    9072*    9076*    9079*    9083*    9120     9141     9171     9173     9277*    9949*
CPSPUR  054050       1431     1480     1521     1625     2214     9898#    9943
CPUERR= 177766        176#    1628*    9360*    9900     9951*   10321*   10332*   10350*   10360*   10370*
CR    = 000015         51#    9473     9483    11729
CRLF  = 000200         52#    1287     1295     9446     9483    11353    11355    11362    11372    11375    11382    11392    11402
                    11405    11415    11421    11424    11429    11436    11441    11448    11458    11461    11468    11473    11487
                    11500    11512    11526    11539    11549    11562    11567    11575    11581    11587    11592    11598    11604
                    11617    11629    11634    11640    11650    11656    11661    11667    11678    11690    11712    11718    11720
                    11722    11727    11761    11784    11798    11804    11813    11822    11833    11865    11873    11883    11937
                    11964    11985    12005    12027    12040    12053    12067    12094    12123    12145    12150    12159    12169
                    12193    12216    12240    12256    12281    12473    12482    12666    12682    12699    12708    12726    12733
                    12746    12768    12775    12788    12810    12817    12830    12852    12859    12872    12894    12901    12914
                    12934    12941    12954    12975    12982    12996    13016    13023    13036    13047    13104    13110    13117
                    13129    13160    13184    13201
CVSPE   047112       8751#
CVSPEA  047124       8754#    8850     8858
CYCNT   036066       6723     6752#
DD    = 000024       4773#
DDCNTR  025040       4788*    4804     4811     4818*    4825#
```

```
DDDONE  025042          4808     4813     4827#
DDPD    024710          4787     4794#
DDPER   025006          4784     4815#
DDPER1  025032          4816     4821#
DDPU1   024746          4803#
DDPU2   024772          4805     4810#
DDPV    024722          4795     4797#
DDTMP   025036          4786*    4798     4803     4823#
DD1     024654          4781     4784#
DF1     113117           617    13648#
DF10  = 113131           638    13666#
DF100   113400          1022     1029     1036    13797#
DF103 = 113372           912    13799#
DF104 = 113372           919    13800#
DF105 = 113372           926    13801#
DF106 = 113372           933    13802#
DF107   113406           940    13803#
DF11    113141           641    13668#
DF110 = 113372           947    13806#
DF111 = 113372           954    13807#
DF112 = 113372           961    13808#
DF113 = 113374           968    13809#
DF115 = 113372           980    13810#
DF12    113152           644    13672#
DF123   113413          1011     1016     1041     1047     1053     1059    13811#
DF13    113160           647    13675#
DF136   113416          1067    13812#   13814
DF137 = 113416          1070    13814#
DF14    113165           650    13678#
DF140   113422          1073    13816#   13819    13821    13823    13825    13827    13829    13831
DF141 = 113422          1076    13819#
DF142 = 113422          1079    13821#
DF143 = 113422          1082    13823#
DF144 = 113422          1085    13825#
DF145 = 113422          1088    13827#
DF146 = 113422          1091    13829#
DF147 = 113422          1094    13831#
DF15    113172           653    13681#
DF150   113426          1097    13833#
DF151   113432          1100    13836#   13838    13839
DF152 = 113432          1103    13838#
DF153 = 113432          1106    13839#
DF154   113435          1109    13841#   13844    13845    13846    13847    13848
DF155 = 113435          1112    13844#
DF156 = 113435          1115    13845#
DF157 = 113435         13846#
DF16    113174           656    13683#
DF160 = 113435          1121    13847#
DF161 = 113435          1124    13848#
DF17    113202           659    13686#   13695
DF2     113123           620    13651#   13654    13656
DF20  = 113202           662    13695#
DF21    113232           665    13697#
DF22    113260           668    13706#   13711    13721    13725
DF23    113263           671    13708#   13713    13723    13727
DF24  = 113260           674    13711#
```

```
DF25    = 113263            677   13713#
DF26      113270            680   13715#
DF27      113274            683   13718#
DF3     = 113123            623   13654#
DF30    = 113260            686   13721#
DF31    = 113263            689   13723#
DF32    = 113260            692   13725#
DF33    = 113263            695   13727#
DF34      113301            698   13729#  13732
DF35    = 113301            701   13732#
DF36      113306            704   13734#
DF37      113312            707   13737#
DF4     = 113123            626   13656#
DF40    = 113324            710   13750#
DF41      113320            713   13741#  13744
DF42    = 113320            716   13744#
DF43      113324            719   13746#  13750
DF44      113334            722   13752#  13758   13767   13769
DF45    = 113334            725   13758#
DF46      113353            728   13760#  13765   13771   13773
DF47    = 113353            731   13765#
DF5       113131            629   13658#  13662   13664   13666
DF50    = 113334            734   13767#
DF51    = 113334            737   13769#
DF52    = 113353            740   13771#
DF53    = 113353            743   13773#
DF54      113367            746   13775#
DF55      113372           1128    1132    1136    1140    1144    1148    1152    1156    1160    1164    1168    1172   13776#
                          13777   13778   13779   13780   13781   13782   13783   13784   13799   13800   13801   13802   13806
                          13807   13808   13810
DF56    = 113372          13777#
DF57    = 113372            771   13778#
DF6     = 113131            632   13662#
DF60    = 113372          13779#
DF61    = 113372            779     786   13780#
DF62    = 113372            794   13781#
DF63    = 113372            802   13782#
DF64    = 113372            810   13783#
DF65    = 113372            818   13784#
DF66      113374            826   13785#  13787   13788   13789   13790   13791   13792   13793   13794   13795   13809
DF67    = 113374            834   13787#
DF7     = 113131            635   13664#
DF70    = 113374            842   13788#
DF71    = 113374            849   13789#
DF72    = 113374            856   13790#
DF73    = 113374            863   13791#
DF74    = 113374            870   13792#
DF75    = 113374            877   13793#
DF76    = 113374            884   13794#
DF77    = 113374            891   13795#
DH1       107657            617   13231#
DH10    = 110033            638   13267#
DH107     111765            938   13501#
DH11      110065            641   13269#
DH111     112037            952   13509#
DH115     112054            978   13512#
```

```
DH12     110141          644   13278#
DH123    112073         1009    1014    13516#
DH125    112122         1020    1027     1034    13520#
DH13     110234          647   13290#
DH130    112211         1039    1045    13530#
DH132    112240         1051    1057    13534#
DH136    112267         1067   13540#   13548
DH137 =  112267         1070   13548#
DH14     110307          650   13299#
DH140    112334         1073   13550#   13557    13559    13561    13563    13565    13567    13569
DH141 =  112334         1076   13557#
DH142 =  112334         1079   13559#
DH143 =  112334         1082   13561#
DH144 =  112334         1085   13563#
DH145 =  112334         1088   13565#
DH146 =  112334         1091   13567#
DH147 =  112334         1094   13569#
DH15     110402          653   13311#
DH150    112377         1097   13571#
DH151    112463         1100   13581#
DH152    112532         1103   13589#
DH153    112601         1106   13598#
DH154    112663         1109   13609#
DH155    112721         1112   13616#
DH156    112757         1115   13623#
DH16     110426          656   13316#
DH160    113015         1121   13631#
DH161    113043         1124   13636#
DH162    113073         1128    1132     1136     1140     1144     1148     1152     1156     1160     1164     1168     1172    13641#
DH17     110514          659   13327#   13337
DH2      107732          620   13240#   13253    13255
DH20  =  110514          662   13337#
DH21     110574          665   13339#
DH22     110647          668   13348#   13367    13391    13395
DH23     110716          671   13356#   13369    13393    13397
DH24  =  110647          674   13367#
DH25  =  110716          677   13369#
DH26     111006          680   13371#
DH27     111053          683   13379#
DH3   =  107732          623   13253#
DH30  =  110647          686   13391#
DH31  =  110716          689   13393#
DH32  =  110647          692   13395#
DH33  =  110716          695   13397#
DH34     111145          698   13399#   13409
DH35  =  111145          701   13409#
DH36     111225          704   13411#
DH37     111272          707   13419#
DH4   =  107732          626   13255#
DH40  =  111424          710   13449#
DH41     111357          713   13431#   13439
DH42  =  111357          716   13439#
DH43     111424          719   13441#   13449
DH44     111473          722   13451#   13463    13472    13474
DH45  =  111473          725   13463#
DH46     111566          728   13465#   13470    13476    13478
```

```
DH47   = 111566            731   13470#
DH5      110033            629   13257#  13263   13265   13267
DH50   = 111473            734   13472#
DH51   = 111473            737   13474#
DH52   = 111566            740   13476#
DH53   = 111566            743   13478#
DH54     111612            746   13480#
DH55     111653            752     761     769     777     784     792     800     808     816     910     917     924     931
                           945     959   13486#
DH6    = 110033            632   13263#
DH66     111667            824     832     840     854     861     868     875     882     889     966   13489#
DH7    = 110033            635   13265#
DH71     111733            847   13496#
DISPLA=  177570             46#   9307*   9331*
DMMA   = 004000            429#
DRH4T1   040416           7138*   7152    7155*   7165    7293#
DRH4T2   040420           7139*   7153    7156*   7166    7294#
DRK5T1   041230           7303*   7317    7320*   7330    7462#
DRK5T2   041232           7304*   7318    7321*   7331    7463#
DRP4T1   037624           6970*   6984    6987*   6997    7129#
DRP4T2   037626           6971*   6985    6988*   6998    7130#
DRRH4    037630           6769    7137#
DRRK5    040422           6770    7302#
DRRP4    037016           6768    6969#
DRRS4    036204           6767    6801#
DRS4T1   037012           6802*   6816    6819*   6829    6961#
DRS4T2   037014           6803*   6817    6820*   6830    6962#
DRUBE    041234           6771    7469#
DT     = 000001            438#   8889
DT1      113442            617   13855#
DT10   = 113472            638   13874#
DT100    114534          14018#
DT107    114552            939   14021#
DT11     113514            641   13876#
DT12     113540            644   13881#
DT123    114566           1010    1015    1040    1046    1052    1058   14024#
DT125    114576           1021    1028    1035   14026#
DT13     113556            647   13885#
DT136    114614           1067   14030#  14033
DT137  = 114614           1070   14033#
DT14     113572            650   13888#
DT140    114626           1073   14035#  14038   14040   14042   14044   14046   14048   14050
DT141  = 114626           1076   14038#
DT142  = 114626           1079   14040#
DT143  = 114626           1082   14042#
DT144  = 114626           1085   14044#
DT145  = 114626           1088   14046#
DT146  = 114626           1091   14048#
DT147  = 114626           1094   14050#
DT15     113606            653   13891#
DT150    114640           1097   14052#
DT151    114652           1100   14055#  14058   14059   14064   14065   14066
DT152  = 114652           1103   14058#
DT153  = 114652           1106   14059#
DT154    114662           1109   14060#  14062   14063
DT155  = 114662           1112   14062#
```

```
DT156 = 114662        1115    14063#
DT157 = 114652       14064#
DT16    113614         656    13893#
DT160 = 114652        1121    14065#
DT161 = 114652        1124    14066#
DT162   114674        1128     1132     1136     1140     1144     1148     1152     1156     1160     1164     1168     1172    14067#
DT17    113632         659    13897#
DT2     113454         620    13858#   13862    13864
DT20    113714         662    13907#
DT21    113776         665    13917#
DT22    114054         668    13926#
DT23    114064         671    13929#
DT24    114100         674    13932#
DT25    114110         677    13935#
DT26    114124         680    13938#
DT27    114136         683    13941#
DT3   = 113454         623    13862#
DT30    114152         686    13944#
DT31    114162         689    13947#
DT32    114176         692    13950#
DT33    114206         695    13953#
DT34    114222         698    13956#   13959
DT35  = 114222         701    13959#
DT36    114236         704    13961#
DT37    114250         707    13964#
DT4   = 113454         626    13864#
DT40  = 114276         710    13977#
DT41    114264         713    13968#   13971
DT42  = 114264         716    13971#
DT43    114276         719    13973#   13977
DT44    114320         722    13979#   13986
DT45  = 114320         725    13986#
DT46    114360         728    13988#   13994
DT47  = 114360         731    13994#
DT5     113472         629    13866#   13870    13872    13874
DT50    114412         734    13996#   14003
DT51  = 114412         737    14003#
DT52    114452         740    14005#   14011
DT53  = 114452         743    14011#
DT54    114504         746    14013#
DT55    114514         753      762      770      778      785      793      801      809      817      911      918      925      932
                       946      953      960      979    14015#
DT6   = 113472         632    13870#
DT66    114522         825      833      841      848      855      862      869      876      883      890      967    14016#
DT7   = 113472         635    13872#
DUBET1  041602        7469*    7488     7491*    7495     7559#
DUBET2  041604        7470*    7489     7492*    7560#
DUT   = 000002         437#
EE    = 000022        4441#
EEDONE  023712        4524     4568     4586#
EEERR1  023420        4447     4531#
EEERR2  023446        4535     4537#
EEERR3  023632        4483     4495     4507     4519     4571#
EETMP1  023412        4526#
EETMP2  023414        4473     4527#
EE1     023202        4477#
```

```
EE10      023344           4508    4512#    4567
EE11      023374           4519#   4522
EE12      023402           4518    4521#
EE13      023406           4520    4524#
EE2       023224           4483#   4486
EE3       023232           4482    4485#
EE4       023240           4484    4488#    4559
EE5       023266           4495#   4498
EE6       023274           4494    4497#
EE7       023302           4496    4500#    4563
EE8       023332           4507#   4510
EE9       023340           4506    4509#
EMTVEC= 000030              149#   1268*    1269*
EM1       070420            617   11742#
EM10      071360            638   11833#
EM100     100062          12491#
EM103     100122            909   12497#
EM104     100203            916   12506#
EM105     100307            923   12518#
EM106     100363            930   12526#
EM107     100453            937   12536#
EM11      071451            641   11845#
EM110     100572            944   12550#
EM111     100620            951   12554#
EM112     100702            958   12563#
EM113     100743            965   12569#
EM115     101001            977   12575#
EM12      071537            644   11856#
EM123     101116           1008   12590#
EM124     101164           1013   12597#
EM125     101226           1019   12603#
EM126     101266           1025   12609#
EM127     101417           1032   12624#
EM13      071664            647   11873#
EM130     101546           1038   12640#
EM131     101614           1044   12647#
EM132     101656           1050   12653#
EM133     101724           1056   12660#
EM136     101766           1067   12666#
EM137     102203           1070   12691#
EM14      071744            650   11883#
EM140     102421           1073   12717#
EM141     102762           1076   12759#
EM142     103322           1079   12801#
EM143     103664           1082   12843#
EM144     104225           1085   12885#
EM145     104557           1088   12925#
EM146     105110           1091   12966#
EM147     105443           1094   13007#
EM15      072003            653   11890#
EM150     105775           1097   13047#
EM151     106061           1100   13057#   13067    13068
EM152 = 106061             1103   13067#
EM153 = 106061             1106   13068#
EM154     106142           1109   13070#
EM155     106174           1112   13076#
```

```
EM156     106226         1115    13082#
EM16      072053          656    11898#
EM160     106273         1121    13091#
EM161     106325         1124    13097#
EM162     106373         1128    13104#
EM163     106603         1132    13129#
EM164     106675         1136    13140#
EM165     106754         1140    13148#
EM166     106773         1144    13151#
EM167     107057         1148    13160#
EM17      072127          659    11907#   11917
EM170     107163         1152    13172#
EM171     107226         1156    13178#
EM172     107272         1160    13184#
EM173     107356         1164    13194#
EM174     107544         1168    13215#
EM175     107607         1172    13221#
EM2       070505          620    11752#
EM20    = 072127          662    11917#
EM21      072210          665    11919#
EM22      072274          668    11929#   11973     12016     12036
EM23      072510          671    11955#   11975
EM24    = 072274          674    11973#
EM25    = 072510          677    11975#
EM26      072644          680    11977#
EM27      073011          683    11996#
EM3       070677          623    11776#
EM30    = 072274          686    12016#
EM31      073156          689    12018#   12038
EM32    = 072274          692    12036#
EM33    = 073156          695    12038#
EM34      073315          698    12040#
EM35      073421          701    12053#
EM36      073530          704    12067#
EM37      073662          707    12084#
EM4       071013          626    11790#
EM40      073744          710    12094#
EM41      074117          713    12115#
EM42      074303          716    12137#
EM43      074556          719    12169#
EM435     107424        13201#
EM44      074704          722    12185#   12264
EM45      075100          725    12208#   12266
EM46      075277          728    12232#   12268
EM47      075420          731    12248#   12270
EM5       071126          629    11804#
EM50    = 074704          734    12264#
EM51    = 075100          737    12266#
EM52    = 075277          740    12268#
EM53    = 075420          743    12270#
EM54      075544          746    12272#
EM55      075717          751    12291#
EM56      075750          760    12296#
EM57      076016          768    12303#
EM6       071206          632    11813#
EM60      076057          776    12309#
```

```
EM61    076116         783     12315#
EM62    076205         791     12325#
EM63    076237         799     12330#
EM64    076315         807     12338#
EM65    076376         815     12347#
EM66    076464         823     12356#
EM67    076574         831     12368#
EM7     071266         635     11822#
EM70    076707         839     12381#
EM71    077023         846     12394#
EM72    077071         853     12401#
EM724   070326        1660     11729#
EM73    077220         860     12417#
EM74    077330         867     12430#
EM75    077432         874     12441#
EM76    077546         881     12454#
EM77    077657         888     12467#
ENDKB   005062        1383      1388#
ERRNG   066123        9981     11487#
ERRVEC= 000004         142#     1445*     1480*     1494*     1521*     1574*     1625*     1650*     1695*     2214*     9266      9267*     9269*
                      9272*     9785      9786      9790*     9795*     9807*     9812*     9822*     9834*     9835*     9943*
ERTYPE  054744        3195      5218      9344      9931     10128#
ERT1    055056       10158#    10222
ERT2    055270       10162     10167     10177     10185     10191     10203     10212     10216#
ERT3    055274       10197     10219#
ERT4    055304       10156     10221     10224#
ERT5    055306       10136     10225#
FCAC  = 000400         432#     8002      8031      8039      8061      8073      8127      8330      8452      8551      8552
FF    = 000021        4303#
FFDONE  023024        4363      4407      4425#
FFERR1  022532        4308      4370#
FFERR2  022560        4374      4376#
FFERR3  022744        4322      4334      4346      4358      4410#
FFTMP1  022524        4365#
FFTMP2  022526        4312      4366#
FF1     022302        4316#
FF10    022454        4347      4351#     4406
FF11    022504        4358#     4361
FF12    022512        4357      4360#
FF13    022520        4359      4363#
FF2     022326        4322#     4325
FF3     022334        4321      4324#
FF4     022344        4323      4327#     4398
FF5     022372        4334#     4337
FF6     022400        4333      4336#
FF7     022410        4335      4339#     4402
FF8     022440        4346#     4349
FF9     022446        4345      4348#
FVPE  = 002000         430#     8773      8779      8793      8889
GETBUF  042066        6809      6977      7145      7310      7474      7642#
GETMP1  042064        7637#     7643*     7664
GGFLG1  026660        5029*     5053      5077      5084      5099      5108      5119*     5123#
GGGM    026664        5031*     5038      5058      5118*     5126#
GGGS    026662        5030*     5036      5041      5117*     5125#
GG1     026210        5029#
GG10    026572        5098      5104#
```

```
GG11    026626           5107      5114#
GG12    026666           5120      5128#
GG2     026232           5034#     5121
GG3     026242           5036#     5040
GG4     026320           5046      5047#     5057
GG5     026412           5064      5065#     5068
GG6     026440           5072      5073#     5091
GG7     026470           5076      5081#
GG8     026520           5082      5090#
GG9     026542           5094      5095#     5115
GNS   = ****** U          475      1286      1294      9695      9696      9697      9698      9699      9700      9701      9703      9704      9705
                         9706      9707      9708      9709      9710      9711      9712      9714      9715      9716      9717      9718
GTBIHI  041772           7620#     7622*     7623
GTBILO  041770           7619#     7621*     7624
GTBINT  041752           6689      7613#
GTMSIZ  042054           7626*     7633#
GTRNH   042062           7630*     7636#     7648      7652*
GTRNL   042060           7629*     7635#     7647      7650*
G1GOV   044652           8218#     8227
G1GOVA  044656           8220#     8275
G1GOVB  044666           8222#     8281
HH    = 000023           4598#
HHDONE  024612           4684      4728      4746#
HHERR1  024320           4606      4691#
HHERR2  024346           4695      4697#
HHERR3  024532           4643      4655      4667      4679      4731#
HHTMP1  024312           4686#
HHTMP2  024314           4633      4687#
HH1     024070           4637#
HH10    024242           4668      4672#     4727
HH11    024272           4679#     4682
HH12    024300           4678      4681#
HH13    024306           4680      4684#
HH2     024114           4643#     4646
HH3     024122           4642      4645#
HH4     024132           4644      4648#     4719
HH5     024160           4655#     4658
HH6     024166           4654      4657#
HH7     024176           4656      4660#     4723
HH8     024226           4667#     4670
HH9     024234           4666      4669#
HIADRS= 177742            161#     1934      2164      2362      2559      3252      3820      4118      4246      4381      4542      4702      5264
                         5825      9916
HIAFLG  054442           9970     10008#
HIAFL2  054456          10014#
HIMFLG  054452           9998     10012#
HIMFL2  054466          10018#
HITMIS= 177752            165#      447#     2956      2976      3181      3435      3697      3768      3995      4068      4885      4901      4917
                         4933      5051      5075      5097      5169      5193      5337      5383      5491      5537      5659      5923      5937
                         5952      5966      6026      6045      6065      6084      6251      6265      6280      6294      6327      6344      6363
                         6380      6399      6416      6435      6452      6477      6500      6523      6546      7710      8144      8180      8249
                         8347      8383      8454      8504      8556      8558      8623      8649      8711      8726      8895      9118      9139
HMRNG   066433          10002     11526#
HT    = 000011             49#     9444      9483
IIA   = 000031           5316#
IIAR1   030204           5331      5424#
```

```
IIAR2   030244          5374    5434#
IIAT1   030202          5332    5370    5378    5416    5422#   5428    5438
IIA1    027702          5325#   5447
IIA2    027726          5328    5330#
IIA3    030026          5367#
IIA4    030046          5374#   5375    5433
IIA5    030160          5413#
IIA6    030200          5420#
IIA7    030300          5329    5420    5444#
IIA8    030316          5445    5449#
IIB   = 000032          5470#
IIBR1   030660          5485    5578#
IIBR2   030720          5528    5588#
IIBT1   030656          5486    5524    5532    5570    5576#   5582    5592
IIB1    030354          5479#   5601
IIB2    030400          5482    5484#
IIB3    030502          5521#
IIB4    030522          5528#   5529    5587
IIB5    030634          5567#
IIB6    030654          5574#
IIB7    030754          5483    5574    5598#
IIB8    030772          5599    5603#
INDONE  042204          6794    7671#
INTMP1  036064          6714*   6735    6749#
INTO    035656          6689#
INT1    035674          6696#
INT2    035720          6702#   6741
INT3    035746          6712#   6719
INT4    035764          6713    6717#
INT5    036000          6716    6723#
INT6    036046          6705    6721    6737#
IOTVEC= 000020           147#   1266*   1267*
IVFC    045622          8419#   8428    8478
IVFCA   045634          8422#   8527    8533
IVSS  = 040000           426#   7933    7934    7939    7973    8060    8447
JJCNT   026130          4845*   4979*   4981    4987    4993    4999#
JJPAT1  026132          4856*   4866    4949    4978    4983*   4989*   4995*   5001#  13866
JJPAT2  026134          4854    4861*   4867    4955    4965    4966    4983    4984*   5002#  13866
JJPAT3  026136          4863*   4870    4871    4872    4891    4907    4989    4990*   5003#  13866
JJPAT4  026140          4864*   4875    4876    4877    4923    4939    4961    4995    4996*   5004#  13866
JJPAT5  026142          4964*   4967    4970*   4972    4978*   4984    4990    4996    5005#
JJTMP1  026144          5007#
JJTMP2  026146          4846    5008#
JJ1     025200          4865#   4976    4985    4991    4997
JJ10    025646          4945    4946#
JJ11    025702          4953    4954#
JJ12    025722          4956    4960#
JJ13    025756          4962    4970#
JJ14    025772          4968    4974#
JJ15    026050          4982    4987#
JJ16    026100          4988    4993#
JJ17    026156          4994    5012#
JJ2     025306          4882#
JJ3     025346          4886    4891#
JJ4     025376          4897    4898#
JJ5     025436          4902    4907#
```

```
JJ6     025466           4913    4914#
JJ7     025526           4918    4923#
JJ8     025556           4929    4930#
JJ9     025616           4934    4939#
KBTST   004606           1335#
KB11CM  001714            591#   1335*   1382*   1394    7912    8943    8979    9014    9056
KB11E   001712            589#   1336*   1340*   1378    1380*   1392    1400    8945    8981    9016
KB11EM  001713            590#   7914    9058
KDPAR0= 172360            319#
KDPAR1= 172362            320#
KDPAR2= 172364            321#
KDPAR3= 172366            322#
KDPAR4= 172370            323#
KDPAR5= 172372            324#
KDPAR6= 172374            325#
KDPAR7= 172376            326#
KDPDR0= 172320            297#
KDPDR1= 172322            298#
KDPDR2= 172324            299#
KDPDP3= 172326            300#
KDPDR4= 172330            301#
KDPDR5= 172332            302#
KDPDR6= 172334            303#
KDPDR7= 172336            304#   8970
KERSTK= 001100             36#
KIPAR0= 172340            308#   1775    2005    2218    2419    2691    2815    3122    3331    3623    3921    4449    4609
                         6671    9085*   9765    9799   10259
KIPAR1= 172342            309#   9087*
KIPAR2= 172344            310#   9089*
KIPAR3= 172346            311#   9091*
KIPAR4= 172350            312#   9062
KIPAR5= 172352            313#
KIPAR6= 172354            314#   1532*   1667*   1884*   2117*   2310*   2510*   2714*   2729*   2753*   2950*   2970*   3173*
                         3420*   3690*   3763*   3989*   4063*   7594*
KIPAR7= 172356            315#   9093*   9816
KIPDR0= 172300            286#   1360*   1361    1363*   1777    2007    2220    2421    2693    2817    3124    3333    3625
                         3923    4451    4611    6673    8947    9086*   9198
KIPDR1= 172302            287#   9088*
KIPDR2= 172304            288#   9090*
KIPDR3= 172306            289#   9092*
KIPDR4= 172310            290#   9063
KIPDR5= 172312            291#
KIPDR6= 172314            292#   1533*   1605*   1645*
KIPDR7= 172316            293#   9094*
KKCNT1  033530           5907*   6125*   6127    6134    6141    6152#
KKERR1  033556           6036    6165#
KKERR2  033570           6056    6169#
KKERR3  033610           6075    6174#
KKERR4  033624           6095    6178#
KKERR5  033646           6167    6172    6176    6181    6183#
KKFLG1  033526           5882*   5977    6097    6099*   6103*   6149#
KKPAT1  033532           5877*   5984    6015    6021    6124    6130*   6137*   6144*   6155#  13922
KKPAT2  033534           5878*   5903    5988    6011    6040    6110    6111    6130    6131*   6156#  13922
KKPAT3  033536           5879*   5992    6007    6060    6137    6138*   6157#  13922
KKPAT4  033540           5880*   5996    6003    6079    6106    6144    6145*   6158#  13922
KKPAT5  033542           6109*   6112    6115*   6117    6124*   6131    6138    6145    6159#
```

```
KKTMP1  033544           6161#
KKTMP2  033546           5888     6162#
KK1     032164           5876#
KK10    033272           6094     6097#
KK11    033312           6098     6103#
KK12    033354           6107     6115#
KK13    033370           6113     6119#
KK14    033400           6120     6123#
KK15    033446           6128     6134#
KK16    033476           6135     6141#
KK17    033674           6142     6192#
KK2     032224           5888#
KK3     032330           5909     5910#    6100     6121     6132     6139     6146
KK4     032656           5980     6000#
KK5     032766           5999     6019#
KK6     033044           6035     6038#
KK7     033130           6055     6058#
KK8     033206           6074     6077#
KT      047736           7687     7916     8934#
LF    = 000012             50#    9477     9483    11729    12150
LLCNT1  027366           5186*    5206*    5217     5234#
LLERR1  027374           5240#    5277
LLERR2  027600           5200     5280#
LLERR3  027560           5269     5276#
LLFLG1  027354           5145*    5171     5195     5214     5222*    5226#    5242     5260     5286
LLFLG2  027356           5187*    5207*    5211     5227#
LLFLG4  027360           5189*    5204     5229#    5253*    5280*
LLGM    027364           5147*    5153     5176     5220*    5232#
LLGS    027362           5146*    5155     5163     5221*    5231#
LLMER   027370           5236#    5240*    5245     5266
LLTMP1  027372           5238#    5241*    5244*    5245
LL1     026720           5145#
LL10    027646           5223     5294#
LL2     026742           5148#    5150     5224
LL3     026766           5153#    5158
LL4     027136           5182#
LL5     027166           5188     5189#    5209
LL6     027224           5194     5198#
LL7     027234           5199     5202#    5278     5293
LL8     027254           5205     5208#
LL9     027326           5212     5220#
LOADRS= 177740            160#    1909     1911     1933     2140     2142     2163     2337     2339     2361     2534     2536     2558
                         3225     3227     3251     3273     3533     3535     3819     3826     4117     4124     4238     4245     4373
                         4380     4534     4541     4694     4701     5249     5263     5428     5438     5582     5592     5824     9910
                         9915
LOAFLG  054440           9968    10007#
LOAFL2  054454          10013#
LOC   = 030624           1452#    1453#    1454#    1455     1500#    1501#    1502#    1503     1582#    1583#    1584#    1585     1609#
                         1610#    1611#    1612     1677#    1678#    1679#    1680     5354#    5355#    5356#    5357     5403#    5404#
                         5405#    5406     5508#    5509#    5510#    5511     5557#    5558#    5559#    5560
LOOP    005144           1415#    9240
MAINT = 177750            164#    1341*    1344     1449     1497     1535     1578     1604     1670     1894     2127     2321     2521
                         3450     5348     5394     5502     5548     9946*   10045*
MANFLG  054450           9991    10011#
MANFL2  054464          10017#
MAPHO = 170202            403#
```

```
MAPH00= 170202          339#    403    1345    2304*   2306*   2504*   2506*   2713*   2728*   2752*
MAPH01= 170206          341#    405
MAPH02= 170212          343#    407
MAPH03= 170216          345#    409
MAPH04= 170222          347#    411
MAPH05= 170226          349#    413
MAPH06= 170232          351#    415
MAPH07= 170236          353#    417
MAPH1 = 170206          405#
MAPH10= 170242          355#
MAPH11= 170246          357#
MAPH12= 170252          359#
MAPH13= 170256          361#
MAPH14= 170262          363#
MAPH15= 170266          365#
MAPH16= 170272          367#
MAPH17= 170276          369#
MAPH2 = 170212          407#
MAPH20= 170302          371#
MAPH21= 170306          373#
MAPH22= 170312          375#
MAPH23= 170316          377#
MAPH24= 170320          379#
MAPH25= 170326          381#
MAPH26= 170332          383#
MAPH27= 170336          385#
MAPH3 = 170216          409#
MAPH30= 170342          387#
MAPH31= 170346          389#
MAPH32= 170352          391#
MAPH33= 170356          393#
MAPH34= 170362          395#
MAPH35= 170366          397#
MAPH36= 170372          399#
MAPH37= 170376          401#
MAPH4 = 170222          411#
MAPH5 = 170226          413#
MAPH6 = 170232          415#
MAPH7 = 170236          417#
MAPL0 = 170200          402#
MAPL00= 170200          338#    402    2303*   2305*   2503*   2505*   2712*   2727*   2751*   4454    4614    11166
MAPL01= 170204          340#    404
MAPL02= 170210          342#    406
MAPL03= 170214          344#    408
MAPL04= 170220          346#    410
MAPL05= 170224          348#    412
MAPL06= 170230          350#    414
MAPL07= 170234          352#    416
MAPL1 = 170204          404#
MAPL10= 170240          354#
MAPL11= 170244          356#
MAPL12= 170250          358#
MAPL13= 170254          360#
MAPL14= 170260          362#
MAPL15= 170264          364#
MAPL16= 170270          366#
```

```
MAPL17= 170274             368#
MAPL2 = 170210             406#     1648
MAPL20= 170300             370#    11025
MAPL21= 170304             372#
MAPL22= 170310             374#
MAPL23= 170314             376#
MAPL24= 170320             378#
MAPL25= 170324             380#
MAPL26= 170330             382#
MAPL27= 170334             384#
MAPL3 = 170214             408#
MAPL30= 170340             386#
MAPL31= 170344             388#
MAPL32= 170350             390#
MAPL33= 170354             392#
MAPL34= 170360             394#
MAPL35= 170364             396#
MAPL36= 170370             398#
MAPL37= 170374             400#
MAPL4 = 170220             410#
MAPL5 = 170224             412#
MAPL6 = 170230             414#
MAPL7 = 170234             416#
MEMERR= 177744             162#     1466*     1470*     1474*     1479*     1515*     1520*     1627*     1701*     1928*     1931      1938*     2159*
                          2161      2168*     2357*     2360      2366*     2554*     2557      2563*     3242      3250      3278*     3523      3550*
                          3808      3816      3832*     4106      4114      4130*     4235      4247      4258*     4370      4382      4393*     4531
                          4543      4554*     4691      4703      4714*     4815      4817*     5240      5276*     5425      5432*     5435      5442*
                          5579      5586*     5589      5596*     5823      9359*     9843      9846      9849*     9907      9914      9950*
MFPT  = 000007             595#     1338
MFPTTR  005054            1337      1385#
MMDES   054646            9705     10088#
MMERR1  035420            6336      6353      6372      6389      6573#
MMERR2  035432            6408      6425      6444      6461 .    6577#
MMERR3  035444            6575      6580#
MMERR4  035462            6490      6513      6586#
MMERR5  035502            6536      6559      6591#
MMERR6  035522            6589      6595#
MMESRS  064772           10066     11355#
MMPAT1  035376            6217*     6319      6488      6563#    13903     13913
MMPAT2  035400            6218*     6511      6564#    13903
MMPAT3  035402            6219*     6534      6565#    13903     13913
MMPAT4  035404            6220*     6557      6566#    13903     13913
MMRFLG  054444            9977     10009#
MMRFL2  054460           10015#
MMR0  = 177572             187#      191      1544*     1548*     1554*     1561*     1565*     1571*     1607*     1621*     1675*     1688*     1694*
                          1700*     1790*     1887*     1940*     2020*     2120*     2175*     2234*     2314*     2367*     2514*     2570*     2707*
                          2890*     2999*     3138*     3357*     3639*     3937*     4469*     4629*     6686*     9114*     9125*     9131*     9146*
                          9161*     9178*     9186*     9192*     9947*    10248
MMR1  = 177574             188#      192
MMR2  = 177576             189#      193
MMR3  = 172516             190#      194      1646*     1702*     1789*     1886*     1941*     2019*     2119*     2176*     2233*     2312*     2368*
                          2512*     2571*     2706*     2889*     3000*     3137*     3356*     3638*     3936*     4468*     4628*     6687*     9113*
                          9193*     9948*
MMSKIP= 104422            1774      1999      2216      2417      2677      2809      3090      3329      3598      3896      4446      4604      9705#
MMTMP1  035406            6568#
MMTMP2  035410            6222      6569#
```

```
MMVEC = 000250            155#   1522*   1540*   1541*   1557*   1600*
MM1     033730           6217#
MM10    034402           6335    6337#
MM11    034440           6345    6350#
MM12    034456           6352    6354#
MM13    034520           6364    6369#
MM14    034536           6371    6373#
MM15    034574           6381    6386#
MM16    034612           6388    6390#
MM17    034654           6400    6405#
MM18    034672           6407    6409#
MM19    034730           6417    6422#
MM2     034046           6236    6238#
MM20    034746           6424    6426#
MM21    035010           6436    6441#
MM22    035026           6443    6445#
MM23    035064           6453    6458#
MM24    035102           6460    6462#
MM25    035164           6478    6488#
MM26    035176           6489    6493#
MM27    035236           6501    6511#
MM28    035250           6512    6516#
MM29    035310           6524    6534#
MM3     034132           6252    6260#
MM30    035322           6535    6539#
MM31    035362           6547    6557#
MM32    035540           6558    6561    6601#
MM4     034170           6266    6275#
MM5     034216           6281    6289#
MM6     034262           6295    6304    6305#
MM7     034272           6307#   6309
MM8     034304           6312#   6314
MM9     034364           6328    6333#
MNRNG   066325           9995   11512#
MONF    054644           1415*  10059*  10076#
MONTTY  054642           1418*  10067   10074#
MSER  = 177744            448#   8749    8750*   8813    8818    8836    8837*   8838    8840    8898    8920*
MSG1    070203           1391   11712#
MSG2    070242           1404   11718#
MSG3    070253           1396   11720#
MSG4    070265           1398   11722#
MSG5    070316           1402   11727#
MSIZER  054670           9706   10104#
MS01    070007           1308   11690#
MS02    070144           1309   11706#
MS03    070173           1315   11710#
MTA101  066714          11562#
MTA11   065162          11382#  13876
MTA120  067021          11575#
MTA124  067302          11617#
MTA126  067374          11629#
MTA131  067454          11640#
MTA134  067617          11661#
MTA135  067653          11667#
MTA17   065227          11390#  11413   13900   13910
MTA20   065263          11399#  13907
```

```
MTA21   065272           11402#  13917
MTA43   065357           11415#  13973
MTA45   065432           11424#  13979   13988
MTA5    065100           11372#  13866
MTA50   065510           11436#  13996   14005
MTA77   066536           11539#
MTB120  067051           11581#
MTB126  067422           11634#
MTB131  067536           11650#
MTB135  067703           11673#
MTB17   065234           11392#  13897   13907
MTB21 = 065227           11413#  13919
MTB45   065460           11429#  13979   13982   13988   13991   13996   13999   14005   14008
MTB77   066552           11542#
MTC120  067101           11587#
MTC131  067571           11656#
MTC135  067725           11678#
MTC17   065254           11396#  13897
MTC45   065475           11433#  13982   13988   13991   13999   14005   14008
MTC77   066614           11549#
MTD120  067131           11592#
MTD135  067761           11684#
MTD77   066651           11555#
MTE120  067161           11598#
MTF120  067211           11604#
MTG120  067241           11610#
MO    = 000004             436#   8767
MOM1  = 000014             444#    463#   1800    2027    2242    2439    4177    4310    4471    4631    5876    5983    5987
                          5991    5995    6002    6006    6010    6014    6020    6039    6059    6078    6104    6185    8122
                          8135    8140    8167    8172    8234    8235    8325    8338    8343    8370    8375    8436    8437
                          8480    8481    8489    8490    8522    8523    8608    8609    8671    8672    8708    8888    9076
M1    = 000010             435#   8769
M1MO  = 000014             462#    463    2606    2678    4947
NNDEV   043064            7696    7849#
NNDONE  043332            7737    7893    7908#
NNDO    043102            7855#
NND2    043162            7853    7873#
NND3    043246            7875    7891#
NNGRM   042742            7699*   7734*   7811#   7828    7841
NNGRPF  042744            7698*   7713    7730*   7731    7812#
NNGRS   042740            7700*   7733*   7810#   7830    7834
NNRH4   042460            7739#   7895
NNRH4U  042463            7741#   7901*
NNRP4   042644            7720    7785#   7877
NNRP4U  042647            7787#   7883*
NNRS4   042552            7716    7762#   7857
NNRS4U  042555            7764#   7864*
NNSTUP  042754            7702    7823#   7824
NNUD    042736            7703    7716    7720    7808#   7851*   7857*   7877*   7895*
NN1     042254            7696#   7760    7783    7806
NN2     042260            7698#   7865    7884    7902
NN3     042300            7702#   7735
NN5     042420            7719    7723    7725    7730#
NN6     042454            7732    7737#
NOCNC   054624           10051   10071#
OKSIZ   004606            1307    1320#
```

```
PARCNT  054470        5326    5480   10028#
PDMSG1  065536        4791   11441#
PDMSG2  065714        4807   11461#
PIRQ  = 177772          44#   1546*   1549*   1555*   1563*   1566*   1572*
PIRQVE= 000240         154#   1539*   1556*
POWERM  065031        9751   11362#
PP    = 000011        2599#   2663
PPHIAD  012250        2609#   2613
PPLIM   012436        2612*   2615    2617*   2621    2636    2652    2656#
PPLOAD  012246        2608#   2615    2617
PP1     012236        2606#
PP2     012304        2614    2616    2619#
PP3     012324        2625#   2637
PP4     012354        2627    2635#
PP5     012370        2640#   2653
PP6     012424        2644    2651#
PP7     012440        2654    2658#
PR0   = 000000          76#
PR1   = 000040          77#
PR2   = 000100          78#
PR3   = 000140          79#
PR4   = 000200          80#
PR5   = 000240          81#   1576    1593
PR6   = 000300          82#
PR7   = 000340          83#   1540    1602    1622
PS    = 177776          41#     42    1260*   9050    9790
PSW   = 177776          42#   9952*  10597*  10757*  10916*  11102*  11231*
PWRVEC= 000024         148#   1272*   1273*   9724*   9725*   9733*   9748*   9749*
QQERR1  032034        5704    5823#
QQERR2  032112        5831    5834#
QQERR3  032124        5835    5838#
QQERR4  032126        5833    5837    5839#
QQFLG1  031520        5639*   5661    5732#   5812*   5829
QQFLG2  031516        5625*   5730#   5817*
QQGM    031534        5632*   5645    5666    5738#   5810*
QQGS    031532        5631*   5647    5653    5737#   5811*
QQHI    031514        5677*   5686*   5695*   5698*   5719    5724    5727#
QQLO    031512        5676*   5685*   5694*   5697*   5723    5726#
QQPAT1  031522        5621*   5656    5733#   5816*   5834
QQTMP1  031524        5719*   5720*   5721    5734#
QQTMP2  031526        5735#
QQTMP3  031530        5736#
QQ1     031032        5629#   5819
QQ10    031536        5670    5686    5688    5694    5698    5746#
QQ11    031612        5672    5759#
QQ12    031614        5676    5679    5690    5695    5697    5766#
QQ13    031670        5681    5779#
QQ14    031672        5677    5685    5786#
QQ15    031746        5799#
QQ16    031750        5755    5775    5795    5801#   5839
QQ17    031752        5804#
QQ18    032014        5813    5816#
QQ19    032132        5818    5841#
QQ2     031062        5643#   5814
QQ3     031146        5656#   5665
QQ4     031210        5660    5665#
```

```
QQ5     031270           5675     5679#
QQ6     031332           5684     5688#
QQ7     031374           5693     5697#
QQ8     031410           5678     5687     5696     5700#
QQ9     031440           5708     5712#    5808
QQ9.5   031452           5717#
RANDTP  041750           7578*    7581     7610#
RANDWR  041606           7577#    9712
RESMON  054512           1427    10045#   10070
RESREG= 104414           6715     6720     6851     7019     7183     7352     7499     7605     7666     7844     9701#    9881    10276
                        10373    10406    10634    10638    10722    10793    10798    10879    10960    10965    11065    11134    11139
                        11194    11266    11271    11343
RESVEC= 000010            143#    1337*
RHWT    041744           7587*    7591     7601*    7608#
RH4AA   037654           7142#    7274
RH4AA1  040074           7172*    7187#
RH4AA2  040076           7179*    7188#
RH4AA3  040072           7140*    7186#    7190*
RH4AE   004036           1229#   10341*   10342*   11254*
RH4AS   004022           1222#
RH4ASS  036120           6769#
RH4BA   004010           1217#   11253*
RH4BB   040115           7161*    7196#
RH4CC   040116           7167*    7197#
RH4CLR  062750          11291    11342    11346#
RH4CR   036074           6730     6756#    6791     7269*    7281*
RH4CS1  004004           1215#   10335    10341    11263*   11282    11313    11316*   11318    11347
RH4CS2  004014           1219#   10338*   11251*   11281*   11287    11311*   11312*   11315*   11338    11346*
RH4CS3  004040           1230#
RH4CYL  062136          11224#
RH4DA1  062116          11216#   11238*   11255
RH4DA2  062120          11217#   11239*   11305*
RH4DB   004026           1224#
RH4DD   040120           7198#
RH4DFL  056052           7694     7758*   10307*   10378#   10506*
RH4DR   004024           1223#   11255*
RH4DT   004032           1226#   10339
RH4EE   040122           7173*    7199#
RH4ER   004020           1221#   11289    11340
RH4ER1  062076           7277     7749    10301*   11208#   11244*   11290*   11341*
RH4ER2  062100           7282     7753    11209#   11245*   11287*   11338*
RH4ER3  062102           7284     7754    11210#   11246*   11288*   11339*
RH4ER4  062104           7283     7755    11211#   11289*   11340*
RH4FF   040124           7180*    7200#
RH4FLG  062074          10296*   11207#   11226    11262*   11270*   11274*
RH4FT   042750           7694*    7759*    7818#    7892     7900*    7907*
RH4FUN  062112          11214#   11236*   11261
RH4GG   040146           7202     7210#
RH4HAN  062140           9716    11226#
RH4HH   040161           7162*    7216#
RH4H1   062152          11227    11231#
RH4H2   062266          11251#
RH4H3   062376          11265    11269#
RH4H4   062412          11259    11274#
RH4H5   062426          11269    11275    11280#
RH4H51  062430          11281#   11296    11298
```

```
RH4H6    062514          11284    11286    11295#
RH4II    040162           7169*    7217#
RH4JJ    040164           7218#
RH4KK    040166           7174*    7219#
RH4LL    040170           7181*    7220#
RH4MA1   062122          11218#   11240*   11253
RH4MA2   062124          11219#   11241*   11254
RH4MM    040212           7222     7229#
RH4MR1   004030           1225#   11256*
RH4MR2   004012           1218#
RH4NN    040225           7163*    7236#
RH4OO    040226           7170*    7237#
RH4PP    040230           7238#
RH4QQ    040232           7175*    7239#
RH4RB    036132           6775#    7146     7147
RH4RDY   062556          11248    11309#
RH4REG   004002           1214#   10337
RH4REX   004034           1228#   10344
RH4RR    040234           7177*    7240#
RH4SEC   062134          11223#
RH4SS    040320           7260     7266#
RH4ST    004016           1220#   11288    11297    11324    11339
RH4SUN   036106           6762#    7159
RH4S1    062532          11247    11302#
RH4S2    062542          11249    11305#
RH4TMP   062110          11213#   11232*   11235
RH4TRK   062132          11222#
RH4UNI   062114          11215#   11237*   11251    11281    11302*   11310    11312    11315
RH4USE   062106          11212#   11309*   11310*   11311    11346
RH4V     004104           1253#   11258
RH4VEC   062130          11221#   11243*   11276
RH4WC    004006           1216#   11252*
RH4WCT   062126          11220#   11242*   11252    11306*
RH4XX    040336           7270     7273#
RH4YY    040344           7205     7211     7225     7230     7245     7249     7262     7267     7276#
RH4ZZ    040414           7278     7291#
RH4111   040256           7242     7249#
RH4112   040267           7160*    7254#
RH4113   040270           7168*    7255#
RH4114   040272           7256#
RH4115   040274           7176*    7257#
RH4116   040276           7178*    7258#
RK5AA    040446           7307#    7443
RK5AA1   040706           7341*    7356#
RK5AA2   040710           7348*    7357#
RK5AA3   040704           7305*    7355#    7359*
RK5ASS   036122           6770#
RK5BA    004054           1237#   10947*
RK5BB    040727           7326*    7365#
RK5CC    040730           7332*    7366#
RK5CLR   061320          10982    11064    11068#
RK5CR    036076           6757#    6783     7438*    7450*
RK5CS1   004050           1235#   10944*   10948*   10955*   10976    11037*   11038    11042*   11044    11069*   11070
RK5CYL   060370          10909#   11007*
RK5DA    004056           1238#   10945*   10949*   11036*   11041*   11068*
RK5DA1   060350          10901#   10923*   10949    10999    11017*
```

```
RK5DA2   060352          10902#   10924*
RK5DB    004060           1239#
RK5DD    040732           7336*    7367#
RK5DFL   056053          10308*   10379#   10493*
RK5DS    004044           1233#   10353    10980    10988    11049    11062
RK5EE    040734           7342*    7368#
RK5ER    004046           1234#   10979    11052    11061
RK5ER1   060330           7446    10302*   10486    10893#   10930*   10981*   11063*
RK5ER2   060332           7451    10894#   10931*   10979*   11061*
RK5ER3   060334           7453    10895#   10932*   10980*   11062*
RK5ER4   060336           7452    10896#
RK5FF    040736           7349*    7369#
RK5FLG   060326          10297*   10892#   10911    10954*   10964*   10969*
RK5FT    042751           7819#
RK5FUN   060344          10899#   10921*   10953
RK5GG    040760           7371     7379#
RK5HAN   060372           9717    10911#
RK5HH    040773           7327*    7385#
RK5H1    060404          10912    10916#
RK5H2    060524          10944#
RK5H3    060630          10959    10963#
RK5H4    060644          10951    10968#
RK5H5    060672          10963    10968    10975#
RK5H51   060674          10976#   10987    10989
RK5H6    060736          10978    10986#
RK5II    040774           7334*    7386#
RK5JJ    040776           7337*    7387#
RK5KK    041000           7343*    7388#
RK5LL    041002           7350*    7389#
RK5MA1   060354          10903#   10925*   10947    11022    11033*
RK5MA2   060356          10904#   10926*   10948    11023    11032*
RK5MM    041024           7391     7398#
RK5NN    041037           7328*    7405#
RK5OO    041040           7335*    7406#
RK5PP    041042           7339*    7407#
RK5QQ    041044           7344*    7408#
RK5RB    036134           6776#    7311     7312
RK5RDY   061160          10935    11036#
RK5REG   004042           1232#   10355
RK5RR    041046           7346*    7409#
RK5SEC   060366          10908#   11008*
RK5SS    041132           7429     7435#
RK5SUN   036110           6763#    7324
RK5S1    060754          10934    10993#
RK5S2    060776          10937    10999#
RK5S3    061100          10940    11020#
RK5TMP   060342          10898#   10917*   10920
RK5TRK   060364          10907#   11009*
RK5UNI   060346          10900#   10922*   10945    10993    10996*   11036    11041
RK5USE   060340          10897#
RK5V     004106           1254#   10950    10970*
RK5VEC   060362          10906#   10928*   10972
RK5WC    004052           1236#   10946*
RK5WCT   060360          10905#   10927*   10946    11020*
RK5XX    041150           7439     7442#
RK5YY    041156           7374     7380     7394     7399     7414     7418     7431     7436     7445#
```

```
RK5ZZ    041226         7447     7460#
RK5111   041070         7411     7418#
RK5112   041101         7325*    7423#
RK5113   041102         7333*    7424#
RK5114   041104         7338*    7425#
RK5115   041106         7345*    7426#
RK5116   041110         7347*    7427#
RLWT     041746         7586*    7592     7595     7600*    7609#
RP4AA    037042         6974#    7110
RP4AA1   037302         7008*    7023#
RP4AA2   037304         7015*    7024#
RP4AA3   037300         6972*    7022#    7026*
RP4AS    003744         1198#
RP4ASS   036116         6768#
RP4BA    003732         1193#   10622*
RP4BAE   003776         1211#   10623*
RF4BB    037323         6993*    7032#
RP4CC    037324         6999*    7033#
RP4CCC   003764         1206#
RP4CLR   057354        10657    10721    10725#
RP4CR    036072         6727     6755#    6787     7105*    7117*
RP4CS1   003726         1191#   10325    10631*   10648    10685    10688*   10689    10726
RP4CS2   003736         1195#   10620*   10647*   10653    10683*   10684*   10687*   10717    10725*
RP4CS3   004000         1212#
RP4CYL   056464        10591#
RP4DA    003734         1194#   10624*
RP4DA1   056444        10583#   10604*   10625    10670    10673*
RP4DA2   056446        10584#   10605*   10624    10674*
RP4DB    003750         1200#
RP4DC    003762         1205#   10625*
RP4DD    037326         7003*    7034#
RP4DFL   056051         7693     7804*   10306*   10377#   10466*
RP4DS    003740         1196#   10654    10662    10695    10718
RP4DT    003754         1202#
RP4EC1   003772         1209#
RP4EC2   003774         1210#
RP4EE    037330         7009*    7035#
RP4ER1   056424         7113     7795    10300*   10459    10575#   10610*   10656*   10720*
RP4ER2   056426         7118     7799    10576#   10611*   10653*   10717*
RP4ER3   056430         7120     7800    10577#   10612*   10654*   10718*
RP4ER4   056432         7119     7801    10578#   10655*   10719*
RP4FF    037332         7016*    7036#
RP4FLG   056422        10295*   10574#   10593    10630*   10637*   10641*
RP4FT    042747         7693*    7805*    7817#    7874     7889*
RP4FUN   056440        10581#   10602*   10629
RP4GG    037354         7038     7046#
RP4HAN   056466         9715    10593#
RP4HH    037367         6994*    7052#
RP4H1    056500        10594    10597#
RP4H2    056620        10620#
RP4H3    056726        10633    10636#
RP4H4    056742        10627    10641#
RP4H5    056756        10636    10642    10646#
RP4H51   056760        10647#   10661    10663
RP4H6    057044        10650    10652    10660#
RP4II    037370         7001*    7053#
```

```
RP4JJ   037372           7004*   7054#
RP4KK   037374           7010*   7055#
RP4LA   003746           1199#
RP4LL   037376           7017*   7056#
RP4MA1  056450          10585#  10606*  10622
RP4MA2  056452          10586#  10607*  10623   10678*
RP4MM   037420           7058    7065#
RP4MR   003752           1201#
RP4NN   037433           6995*   7072#
RP4OF   003760           1204#  10712*
RP4OO   037434           7002*   7073#
RP4PP   037436           7006*   7074#
RP4QQ   037440           7011*   7075#
RP4RB   036130           6774#   6978    6979
RP4RDY  057132          10615   10681#
RP4REG  003724           1190#  10327
RP4RR   037442           7013*   7076#
RP4RR1  003742           1197#  10655   10719
RP4RR2  003766           1207#
RP4RR3  003770           1208#
RP4SEC  056462          10590#
RP4SN   003756           1203#
RP4SS   037526           7096    7102#
RP4SUN  036104           6761#   6991
RP4S1   057062          10614   10667#
RP4S2   057072          10616   10670#
RP4S3   057116          10618   10677#
RP4TMP  056436          10580#  10598*  10601
RP4TRK  056460          10589#
RP4UNI  056442          10582#  10603*  10620   10647   10667*  10682   10684   10687
RP4USE  056434          10579#  10681*  10682*  10683   10725
RP4V    004102           1252#  10626
RP4VEC  056456          10588#  10609*  10643
RP4WC   003730           1192#  10621*
RP4WCT  056454          10587#  10608*  10621   10677*
RP4XX   037544           7106    7109#
RP4YY   037552           7041    7047    7061    7066    7081    7085    7098    7103    7112#
RP4ZZ   037622           7114    7127#
RP4111  037464           7078    7085#
RP4112  037475           6992*   7090#
RP4113  037476           7000*   7091#
RP4114  037500           7005*   7092#
RP4115  037502           7012*   7093#
RP4116  037504           7014*   7094#
RR    = 000007           2208#
RRADR1  011300           2246*   2247*   2252    2254    2271    2272    2380*   2381*   2387#
RRADR2  011304           2271*   2272*   2273*   2274*   2275*   2283    2285    2303    2304    2337    2339    2389#  13944
                        13947
RRADR3  011310           2244*   2245*   2252    2254    2273    2275    2369    2371    2373*   2376*   2377*   2382*   2383*
                         2391#
RRDONE  011314           2291    2294    2394#
RRHIAD  010550           2238#
RRLOAD  010546           2237#   2240*   2283    2285
RR1     010610           2249#   2385
RR10    011242           2370    2372    2376#
RR11    011254           2260    2265    2380#
```

```
RR12      011274                2374      2378      2385#
RR2       010646                2263      2269#
RR3       010702                2280#
RR4       010740                2296      2299#
RR5       011064                2328#
RR6       011072                2323      2334#
RR7       011130                2345      2354#
RR8       011150                2348      2350      2360#
RR9       011206                2331      2358      2367#
RSET    = 104416                1966      2191      2394      2586      2658      2776      3068      3282      3562      3857      4153      4290      4425
                                4586      4746      4827      5449      5603      6192      6668      7671      7689      7908      9703#     9932     10047
                               10092
RS4AA     036230                6806#     6942
RS4AA1    036470                6840*     6855#
RS4AA2    036472                6847*     6856#
RS4AA3    036466                6804*     6854#     6858*
RS4AS     003706                1182#
RS4ASS    036114                6699      6766      6767#
RS4BA     003674                1177#    10782*
RS4BAE    003720                1187#    10783*
RS4BB     036511                6825*     6864#
RS4CC     036512                6831*     6865#
RS4CLR    060310               10817     10878     10882#
RS4CR     036070                6697      6724      6754#     6789      6937*     6949*
RS4CS1    003670                1175#    10314     10790*    10808     10851     10854*    10855     10882*    10883
RS4CS2    003700                1179#    10780*    10807*    10813     10849*    10850*    10853*    10874
RS4CS3    003722                1188#
RS4CYL    057434               10751#    10836*
RS4DA     003676                1178#    10784*
RS4DA1    057414               10743#    10764*    10784     10830     10841*
RS4DA2    057416               10744#    10765*
RS4DB     003712                1184#
RS4DD     036514                6835*     6866#
RS4DFL    056050                6696      7692      7781*    10305*    10376#    10438*
RS4DS     003702                1180#    10814     10822     10861     10875
RS4DT     003716                1186#
RS4EE     036516                6841*     6867#
RS4ER     003704                1181#    10815     10876
RS4ER1    057374                6945      7772     10299*    10431     10735#    10770*    10816*    10877*
RS4ER2    057376                6950      7776     10736#    10771*    10813*    10874*
RS4ER3    057400                6952      7777     10737#    10772*    10814*    10875*
RS4ER4    057402                6951      7778     10738#    10815*    10876*
RS4FF     036520                6848*     6868#
RS4FLG    057372               10294*    10734#    10753     10789*    10797*    10801*
RS4FT     042746                7692*     7782*     7816#     7852      7863*     7871*     7882*
RS4FUN    057410               10741#    10762*    10788
RS4GG     036542                6870      6878#
RS4HAN    057436                9714     10753#
RS4HH     036555                6826*     6884#
RS4H1     057450               10754     10757#
RS4H2     057570               10780#
RS4H3     057670               10792     10796#
RS4H4     057704               10786     10801#
RS4H5     057720               10796     10802     10806#
RS4H51    057722               10807#    10821     10823
RS4H6     060006               10810     10812     10820#
```

```
RS4II   036556            6833*   6885#
RS4JJ   036560            6836*   6886#
RS4KK   036562            6842*   6887#
RS4LA   003710            1183#
RS4LL   036564            6849*   6888#
RS4MA1  057420           10745#  10766*  10782
RS4MA2  057422           10746#  10767*  10783   10845*
RS4MM   036606            6890    6897#
RS4MR   003714            1185#
RS4NN   036621            6827*   6904#
RS4OO   036622            6834*   6905#
RS4PP   036624            6838*   6906#
RS4QQ   036626            6843*   6907#
RS4RB   036126            6773#   6810    6811    7613    7656
RS4RDY  060116           10775   10847#
RS4REG  003666            1174#  10316
RS4RR   036630            6845*   6908#
RS4SEC  057432           10750#
RS4SS   036714            6928    6934#
RS4SUN  036102            6698    6760#   6823
RS4S1   060024           10774   10827#
RS4S2   060034           10777   10830#
RS4S3   060102           10778   10844#
RS4TMP  057406           10740#  10758*  10761
RS4TRK  057430           10749#  10835*
RS4UNI  057412           10742#  10763*  10780   10807   10827*  10848   10850   10853
RS4USE  057404           10739#  10847*  10848*  10849   10882
RS4V    004100            1251#  10785
RS4VEC  057426           10748#  10769*  10803
RS4WC   003672            1176#  10781*
RS4WCT  057424           10747#  10768*  10781   10844*
RS4XX   036732            6938    6941#
RS4YY   036740            6873    6879    6893    6898    6913    6917    6930    6935    6944#
RS4ZZ   037010            6946    6959#
RS4111  036652            6910    6917#
RS4112  036663            6824*   6922#
RS4113  036664            6832*   6923#
RS4114  036666            6837*   6924#
RS4115  036670            6844*   6925#
RS4116  036672            6846*   6926#
SAVREG= 104412            6708    6808    6976    7144    7309    7473    7580    7646    7823    9700#   9866   10242   10304
                         10399   10600   10760   10919   11105   11234
SDPAR0= 172260             275#
SDPAR1= 172262             276#
SDPAR2= 172264             277#
SDPAR3= 172266             278#
SDPAR4= 172270             279#
SDPAR5= 172272             280#
SDPAR6= 172274             281#
SDPAR7= 172276             282#
SDPDR0= 172220             253#
SDPDR1= 172222             254#
SDPDR2= 172224             255#
SDPDR3= 172226             256#
SDPDR4= 172230             257#
SDPDR5= 172232             258#
```

```
SDPDR6= 172234           259#
SDPDR7= 172236           260#    9006
SETBLE= 036114          6766#
SETMP   056126         10397*   10400    10409#
SETREG  056070         10315    10326    10336    10343    10354    10364    10397#
SIPAR0= 172240           264#
SIPAR1= 172242           265#
SIPAR2= 172244           266#
SIPAR3= 172246           267#
SIPAR4= 172250           268#
SIPAR5= 172252           269#
SIPAR6= 172254           270#
SIPAR7= 172256           271#
SIPDR0= 172200           242#    8983
SIPDR1= 172202           243#
SIPDR2= 172204           244#
SIPDR3= 172206           245#
SIPDR4= 172210           246#
SIPDR5= 172212           247#
SIPDR6= 172214           248#
SIPDR7= 172216           249#
SIZDEV  055456          6691     7691    10294#
SIZE  = 104424          1791     2021     2236     2434     2607     2679     2831     3092     3346     3600     3898     7618     9706#
SIZEHI= 177762           173#
SIZELO= 177760           171#    1305     1311
SIZRH4  056412         10346    10506#
SIZRK5  056300         10356    10471#
SIZRP4  056214         10328    10444#
SIZRS4  056130         10317    10416#
SIZTM1  056056         10311*   10372    10383#
SIZTM2  056060         10384#
SIZTM3  056062         10385#
SIZTM4  056064         10386#
SIZTM5  056066         10387#
SIZUBE  056364         10366    10498#
SKAD    054230          1766*    1994*    2210*    2413*    2601*    2673*    2804*    3086*    3325*    3593*    3889*    4172*    4305*
                        4443*    4600*    4775*    4782     4840*    5024*    5140*    5318*    5472*    5615*    5869*    6211*    6666*
                        7687*    9933     9935#   10073    10093
SKBADR  054322          9707     9968#
SKBCNR  054364          9709     9984#
SKBERR  054346          9708     9977#
SKBHMR  054420          9711     9998#
SKBMNR  054402          9710     9991#
SKIPT = 104420          5344     5390     5498     5544     9704#    9903     9921    10005
SKPBAD= 104426          9707#
SKPBCN= 104432          9709#
SKPBER= 104430          9708#
SKPBHM= 104436          9711#
SKPBMN= 104434          9710#
SKRNG   054434          9975     9982     9989     9996    10004#
SPUR    054076          1432     1522     1626     1769     1997     2213     2416     2604     2676     2807     3100     3236     3238
                        3244     3328     3422     3525     3526     3544     3546     3595     3810     3891     4108     4237     4240
                        4372     4375     4533     4536     4693     4696     4778     4821     4842     5026     5142     5149     5247
                        5251     5321     5427     5437     5475     5581     5591     5618     5874     6213     9906#    9913     9942
SR0   = 177572           191#    9796     9811*    9819*
SR1   = 177574           192#
```

```
SR2    = 177576          193#
SR3    = 172516          194#     9808*
SS     = 000010         2411#
SSADR1   012162         2441*     2442*     2447*     2448*     2449*     2457      2458      2576#
SSADR2   012166         2457*     2458*     2459*     2460*     2463      2465      2482      2484      2503      2504      2534      2536      2578#
                       13950     13953
SSCNST   012176         2463      2465      2583#
SSHIAD   011440         2436#
SSLOAD   011436         2435#     2437*     2482      2484
SSMASK   012172         2451*     2452*     2459      2460      2565*     2566*     2580#
SS1      011456         2441#
SS10     012120         2476      2493      2529      2555      2565#
SS11     012202         2573      2586#
SS2      011476         2446#     2574
SS3      011516         2444      2451#
SS4      011540         2455      2457#     2569
SS5      011626         2471      2474      2479#
SS6      011664         2490      2495      2499#
SS7      012006         2531#
SS8      012044         2523      2542      2551#
SS9      012062         2545      2547      2557#
STACK  = 001500           35#       36        37        38       467#     1265      1450      1498      1543      1560      1595      1606      1647
                        1655      1674      4797      4828      9945
START    004112          479      1259#     9753
STKLMT= 177774            43#
STOP     054602        10061     10065#
SUPSTK= 000700            37#
SWR    = 177570           45#       46      1658      4780      9261*     9275      9278      9286      9293      9332      9335      9342      9346
                        9352     10088
SW0    = 000001          111#
SW00   = 000001          101#      111
SW01   = 000002          100#      110
SW02   = 000004           99#      109
SW03   = 000010           98#      108
SW04   = 000020           97#      107
SW05   = 000040           96#      106
SW06   = 000100           95#      105
SW07   = 000200           94#      104
SW08   = 000400           93#      103
SW09   = 001000           92#      102
SW1    = 000002          110#
SW10   = 002000           91#
SW11   = 004000           90#
SW12   = 010000           89#     4780
SW13   = 020000           88#
SW14   = 040000           87#     1658
SW15   = 100000           86#
SW2    = 000004          109#
SW3    = 000010          108#
SW4    = 000020          107#
SW5    = 000040          106#
SW6    = 000100          105#
SW7    = 000200          104#    10088
SW8    = 000400          103#
SW9    = 001000          102#
SYSTID= 177764           175#
```

```
S0     = 000020           434#    8125     8329     8765     8852     8925     9079
SOMOM1= 000034            443#     461#    3097     3201     3605     3787     3903     4085     4869     5031     5117     5911     6239
                         8108     8220     8279     8311     8420     8531     8592     8680     8703     8752     8856     8876     8929
                         9070
SOM1   = 000030           459#    2841     3005     3359     5147     5221     5330     5484     5632     5811     7700     7734
S1     = 000040           433#    8126     8328     8462     8508     8528     8628     8654     8676     8808     9083
S1MO   = 000044           458#    2842     3004     3493     5146     5220     5376     5530     5631     5810     7699     7733
S1MOM1= 000054            442#     460#    3098     3200     3606     3786     3904     4086     4874     5030     5118     5914     6242
                         8109     8221     8278     8312     8421     8530     8593     8679     8705     8753     8855     8877     8928
                         9072
TAB    = 000011           457#   11370    11375    11382    11396    11399    11405    11415    11424    11433    11436    11562    11567
                        11629    11634    11640    11650    11656    11661    13231    13240    13248    13257    13269    13278    13290
                        13299    13305    13311    13316    13321    13327    13333    13339    13348    13356    13362    13371    13379
                        13399    13411    13419    13424    13431    13441    13451    13457    13465    13480    13540    13550    13571
                        13584    13592    13603    13609    13616    13623    13631    13636
TBITVE= 000014            144#
TESTR1= 140000            464#    4848     5044     5062     5070     5092     5160     5180     5701     5892     6225     6316     6467
                         7706     7744     7767     7790     7837
TESTR2= 142000            465#    4850     5652     5700     5712     5894     6227     6318     6469     7826
TESTR3= 144000            466#    4852
TKVEC  = 000060           151#    1418     1427*    1428*   10067*   10070*
TLOC   = 114704         14069#   14070#   14071#   14072
TMP      045166          8248*    8260     8263     8282#
TOP      005200          1416     1425#
TPVEC  = 000064           152#
TRAPVE= 000034            150#    1270*    1271*
TRTVEC= 000014            145#
TSTDAT   114704          8119     8141     8176     8236     8245     8322     8344     8379     8439     8451     8492     8501     8611
                         8620     8646    14073#
TSTDT1   062764          9077     9099    11349#
TST1     005250          1440#
TST10    011316          2210     2409#
TST11    012204          2413     2597#
TST12    012442          2601     2669#
TST13    013102          2673     2801#
TST14    014254          2804     3081#
TST15    015304          3086     3321#
TST16    016374          3325     3588#
TST17    020016          3593     3884#
TST2     005444          1491#
TST20    021444          3889     4168#
TST21    022230          4172     4301#
TST22    023026          4305     4439#
TST23    023714          4443     4596#
TST24    024614          4600     4772#
TST25    025050          4775     4837#
TST26    026156          4840     5021#
TST27    026666          5024     5137#
TST3     005600          1531#
TST30    027646          5140     5314#
TST31    030320          5318     5468#
TST32    030774          5472     5613#
TST33    032132          5615     5866#
TST34    033676          5869     6208#
TST35    035540          6211     6664#
TST36    042206          6666     7685#
```

```
TST37   043354           7926#
TST4    006300           1644#
TST40   043560           7971#
TST41   043674           7989     8001#
TST42   043754           8027#
TST43   044066           8056#
TST44   044224           8103#
TST45   044650           8198     8217#
TST46   045170           8277     8306#
TST47   045620           8401     8417#
TST5    006576           1762#
TST50   046276           8529     8545#
TST51   046364           8590#
TST52   046730           8677     8699#
TST53   047100           8727     8736     8748#
TST54   047502           8853     8873#
TST55   047736           8941#
TST56   050072           8946     8977#
TST57   050226           8982     9012#
TST6    007504           1766     1990#
TST60   050362           9017     9047#
TST7    010404           1994     2206#
TT    = 000012           2671#
TTHIAD  012510           2681#    2684
TTLIM   013076           2683*    2686     2688*    2720     2744     2770     2774#
TTLOAD  012506           2680#    2686     2688
TT1     012476           2678#
TT2     012544           2685     2687     2689#
TT3     012710           2725#    2745
TT4     012762           2735     2742#
TT5     013002           2749#    2771
TT6     013060           2761     2768#
TT7     013100           2772     2776#
TVADFL  055312          10189*   10201*   10228#   10246
TVADHI  055316          10235#   10245*   10268*
TVADLO  055314          10234#   10244*   10269*   10270
TYPDS = 104410           9226     9699#   10166
TYPE  = 104400           1284     1292     1308     1309     1310     1315     1391     1396     1398     1402     1404     1660     4790
                         4806     9224     9227     9337     9345     9449     9545     9622     9695#    9750     9973     9980     9987
                         9994    10001    10052    10065    10128    10143    10145    10149    10151    10175    10195    10210    10216
                        10274
TYPOC = 104402           9696#   10135    10161
TYPON = 104406           9698#
TYPOS = 104404           1312     1317     9697#   10182
TYPVAD  055320          10190    10202    10242#
T.END   005002           1367     1370#
T1      004634           1341#    1386
T2      004676           1348     1351#
T3      004730           1353     1356     1359#
T4      004762           1362     1365#
UBCLR   062060          11154    11193    11197#
UBEAA   041256           7473#    7544
UBEAA1  041412           7480*    7503#
UBEAA2  041414           7483*    7504#
UBEAA3  041410           7471*    7502#    7506*
UBEASS  036124           6771#
```

```
UBEBA   004070            1245#   11123*
UBEBB   041432            7511#
UBECC   004066            1244#   11122*
UBECCC  041434            7497*    7512#
UBECLR  004074            1247#   11181*   11197*
UBECR   036100            6758#    6785     7540*    7550*
UBECR1  004072            1246#   10498*   10499    11131*   11147    11151    11183    11190    11198
UBECR2  004076            1248#   11124*   11152    11191
UBECYL  061404           11095#
UBEDA1  061364           11087#   11109*
UBEDA2  061366           11088#   11110*   11125
UBEDB   004064            1243#   10363    11125*
UBEDD   041436            7481*    7513#
UBEDFL  056054           10309*   10380#   10501*
UBEEE   041440            7484*    7514#
UBEER1  061344            7546    10303*   11079#   11115*   11153*
UBEER2  061346            7551    11080#   11116*   11151*   11190*
UBEER3  061350            7552    11081#   11117*   11152*   11191*   11192*
UBEER4  061352           11082#
UBEFF   041460            7516     7522#
UBEFLG  061342           10298*   11078#   11097    11130*   11138*   11142*
UBEFT   042752            7820#
UBEFUN  061360           11085#   11108*   11129
UBEGG   041472            7527#
UBEHAN  061406            9718    11097#
UBEHH   041474            7496*    7528#
UBEH1   061420           11098    11102#
UBEH2   061524           11122#
UBEH3   061616           11133    11137#
UBEH4   061632           11127    11142#
UBEH5   061646           11137    11143    11146#
UBEH51  061650           11147#   11159
UBEH6   061712           11149    11158#
UBEII   041476            7482*    7529#
UBEJJ   041500            7485*    7530#
UBEKK   041520            7532     7538#
UBELL   041534            7541     7544#
UBEMA1  061370           11089#   11111*   11123    11163    11176*
UBEMA2  061372           11090#   11112*   11124    11164    11175*
UBERB   036136            6777#    7475     7477
UBERDY  062004           11118    11181#
UBEREG  004062            1242#   10365
UBESEC  061402           11094#
UBESUN  036112            6764#
UBES1   061722           11119    11163#
UBETMP  061356           11084#   11103*   11106
UBETRK  061400           11093#
UBEUNI  061362           11086#
UBEUSE  061354           11083#
UBEV    004110            1255#   11126
UBEVEC  061376           11092#   11114*   11144
UBEWCT  061374           11091#   11113*   11122    11177*   11178*
UBEYY   041540            7518     7522     7534     7538     7546#
UBEZZ   041600            7547     7557#
UCB   = 001000             431#    8618     8645
UDPAR0= 177660             231#
```

```
UDPAR1= 177662            232#
UDPAR2= 177664            233#
UDPAR3= 177666            234#
UDPAR4= 177670            235#
UDPAR5= 177672            236#
UDPAR6= 177674            237#
UDPAR7= 177676            238#
UDPDR0= 177620            209#
UDPDR1= 177622            210#
UDPDR2= 177624            211#
UDPDR3= 177626            212#
UDPDR4= 177630            213#
UDPDR5= 177632            214#
UDPDR6= 177634            215#
UDPDR7= 177636            216#  9041
UIPAR0= 177640            220#
UIPAR1= 177642            221#
UIPAR2= 177644            222#
UIPAR3= 177646            223#
UIPAR4= 177650            224#
UIPAR5= 177652            225#
UIPAR6= 177654            226#
UIPAR7= 177656            227#
UIPDR0= 177600            198#  9018
UIPDR1= 177602            199#
UIPDR2= 177604            200#
UIPDR3= 177606            201#
UIPDR4= 177610            202#
UIPDR5= 177612            203#
UIPDR6= 177614            204#
UIPDR7= 177616            205#
USESTK= 000600             38#
UU    = 000016          3590#   3862
UUADR1  017506          3643*   3644*   3650    3651    3679*   3680*   3703*   3704*   3717*   3718*   3723    3724    3752*
                        3753*   3776*   3777*   3797#
UUADR2  017512          3641*   3642*   3652    3654    3705*   3715*   3716*   3725    3727    3778*   3799#
UUADR3  017516          3650*   3651*   3652*   3653*   3654*   3662    3664    3683    3700    3701    3723*   3724*   3725*
                        3726*   3727*   3735    3737    3756    3772    3773    3801#   3817    3818
UUDONE  020014          3784    3849    3854    3857#
UUERR1  017530          3646    3808#   3835
UUERR2  017544          3809    3812#
UUERR3  017664          3825    3831#
UUERR4  017666          3830    3832#
UUFLG1  017500          3604*   3699    3771    3783    3785*   3790#   3815
UUFLG2  017502          3608*   3647*   3709*   3720*   3782*   3793#   3837    3840    3843    3846    3853
UUFLG3  017504          3597*   3795#   3848    3850*
UUGM    017524          3606*   3614    3711    3787*   3805#
UUGS    017522          3605*   3612    3618    3786*   3804#
UUHIAD  016440          3602#
UULOAD  016436          3601#   3662    3664    3735    3737
UUTMP   017526          3806#   3824*   3834
UU0     016404          3591#   3609
UU1     016462          3608#   3788    3855
UU10    017242          3732#
UU11    017300          3746    3752#
UU12    017314          3743    3748    3756#
```

```
UU13      017406              3769      3776#     3845
UU14      017444              3783#     3851
UU2       016476              3612#     3616
UU3       016676              3648#     3681      3708
UU4       016732              3659#
UU5       016770              3673      3679#
UU6       017004              3670      3675      3683#
UU7       017100              3698      3703#     3839
UU8       017152              3715#     3842
UU9       017206              3721#     3754      3781
VCIP    = 010000               428#     7951      7962      7979      8005      8010      8029      8037      8045      8058      8068      8071      8080
                              8106      8161      8165      8223      8309      8364      8368      8425      8547      8597      8757      8881      9065
VPBP      050422              9057      9059      9061#     9068
VPBPA     050436              9064#     9200
VPBPE     051214              9060      9199      9201#
VSCM      047504              8874#
VSCMA     047516              8878#     8924      8931
VSG0      044226              8104#     8112      8136      8168
VSG0A     044230              8105#     8199
VSG1      045172              8307#     8315      8339      8371
VSG1A     045174              8308#     8402
VSIU    = 020000               427#     7944      7945      7957      7958      7974      7981      7982      7987      7988      8032      8040      8062
                              8070      8074      8130      8163      8196      8270      8333      8366      8399      8524      8673      8830      8848
                              8922
VSPE    = 100000               425#     8800      8904      8913      8914
VV      = 000017              3566      3886#
VVADR1    021134              3941*     3942*     3949      3950      3978*     3979*     4003*     4004*     4017*     4018*     4023      4024      4052*
                              4053*     4076*     4077*     4094#
VVADR2    021140              3939*     3940*     3951      3953      4005*     4015*     4016*     4025      4027      4078*     4096#
VVADR3    021144              3949*     3950*     3951*     3952*     3953*     3961      3963      3982      3999      4000      4023*     4024*     4025*
                              4026*     4027*     4035      4037      4056      4072      4073      4098#     4115      4116
VVDONE    021442              4084      4146      4150      4153#
VVERR1    021156              3944      4106#     4133
VVERR2    021172              4107      4110#
VVERR3    021312              4123      4129#
VVERR4    021314              4128      4130#
VVFLG1    021126              3902*     3998      4071      4083      4087*     4090#     4113
VVFLG2    021130              3906*     3945*     4009*     4020*     4082*     4091#     4134      4137      4140      4143      4149
VVFLG3    021132              3894*     4092#     4145      4147*
VVGM      021152              3904*     3912      4011      4085*     4102#
VVGS      021150              3903*     3910      3916      4086*     4101#
VVHIAD    020062              3900#
VVLOAD    020060              3899#     3961      3963      4035      4037
VVTMP     021154              4104#     4122*     4132
VV0       020026              3887#     3907
VV1       020104              3906#     4088      4151
VV10      020670              4032#
VV11      020726              4046      4052#
VV12      020742              4043      4048      4056#
VV13      021034              4069      4076#     4142
VV14      021072              4083#     4148
VV2       020120              3910#     3914
VV3       020322              3947#     3980      4008
VV4       020356              3958#
VV5       020414              3972      3978#
VV6       020430              3969      3974      3982#
```

```
VV7      020524         3996     4003#    4136
VV8      020576         4015#    4139
VV9      020634         4021#    4054     4081
WAITLP   036140         6694     6782#    6784     6786     6788     6790     6792
WRRAND=  104440         6853     7021     7185     7354     7501     9712#
X      = 000005         1764#    2195
XADR1    007462         1804*    1805*    1810     1812     1829     1830     1952*    1953*    1958#
XADR2    007466         1829*    1830*    1831*    1832*    1833*    1841     1843     1875     1876     1909     1911     1960#   13926
                       13929
XADR3    007472         1802*    1803*    1810     1812     1831     1833     1942     1944     1946*    1948*    1949*    1954*    1955*
                        1962#
XADR4    007476         1964#
XDONE    007502         1849     1852     1966#
XHIADR   006734         1796#
XLOADR   006732         1795#    1797*    1841     1843
XX     = 000006         1992#    2398
XXADR1   010362         2030*    2031*    2036*    2037*    2038*    2046     2047     2181#
XXADR2   010366         2046*    2047*    2048*    2049*    2053     2055     2073     2075     2108     2109     2140     2142     2183#
                       13932    13935
XXCNST   010376         2053     2055     2188#
XXHIA    007642         2023#
XXLOA    007640         2022#    2024*    2073     2075
XXMASK   010372         2040*    2041*    2048     2049     2170*    2171*    2185#
XX1      007660         2030#
XX10     010320         2066     2084     2135     2160     2170#
XX11     010402         2178     2191#
XX2      007700         2035#    2179
XX3      007720         2033     2040#
XX4      007742         2044     2046#    2174
XX5      010030         2061     2064     2070#
XX6      010066         2081     2086     2090#
XX7      010210         2129     2137#
XX8      010246         2148     2157#
XX9      010262         2151     2153     2161#
X1       006774         1807#    1956
X10      007424         1943     1945     1948#
X11      007436         1818     1823     1952#
X12      007456         1947     1950     1956#
X2       007032         1821     1827#
X3       007066         1838#
X4       007124         1854     1857#
X5       007250         1900#
X6       007256         1896     1906#
X7       007314         1917     1926#
X8       007332         1920     1922     1931#
X9       007370         1904     1929     1940#
ZADHI    013224         2833#
ZADLO    013222         2832#    2901     2903     2929     2931
ZCMTBL   014144         2860     2868     2876     3038#
ZFLG1    014020         2835*    2958     2978     3001     3003*    3009#
ZFLG2    014022         2844*    2853     2863     2871     2994*    3006*    3011#
ZGM      014024         2842*    2882     3005*    3017#
ZGS      014026         2841*    2884     3004*    3018#
ZTABLE   014042         2858*    2859*    2866*    2867*    2874*    2875*    2892     2915     3028#    3038     3052     3054     3059
ZTABOT   014142         2987     2991     3029#
ZTHR     014032         2881     3020#    3060
```

```
ZTMP1   014036                3023#     3043*     3050      3051
ZTMP2   014040                3024#     3044*
Z1      013252                2853#     2995      3007
Z10     013530                2934      2937      2943#
Z11     013616                2957      2963#
Z12     013716                2922      2977      2985#
Z13     013730                2939      2990#
Z14     013752                2879      2911      2999#
Z15     014252                3002      3068#
Z2      013300                2854      2863#
Z3      013330                2864      2871#
Z4      013360                2872      2879#
Z5      013364                2861      2869      2877      2881#
Z7      013430                2896#     2992
Z8      013470                2921#     2988
Z9      013474                2924#
$BDADR  001524                 529#
$BDDAT  001530                 531#
$BELL   001702                 585#     9337      9362
$CHARC  052332                9459*     9466      9475*     9480#
$CMTAG  001500                 517#     1260      1261      1268      1274      1275      1276
$CM1  = 000024                 543#      544#      545#      546#      547#      548#      549#      550#      551#      552#      553#      554#      555#
                               556#      557#      558#      559#      560#      561#      562#      563#
$CM2  = 000050                 543#      544#      545#      546#      547#      548#      549#      550#      551#      552#      553#      554#      555#
                               556#      557#      558#      559#      560#      561#      562#      563#
$CM3  = 000024                 541#      543
$CM4  = 000024                 563#      564#      565#      566#      567#      568#      569#      570#      571#      572#      573#      574#      575#
                               576#      577#      578#      579#      580#      581#      582#      583#
$CORE   053566                9794      9822#
$CRLF   001707                 587#     1310      9345      9362      9449      9483     10129     10146     10152     13897     13903     13907     13913
                             13919
$CROUT  053616                9822      9829#
$DBLK   053000                9588      9622      9630#
$DB20   053730                9866#    10172     10208     10271
$DOAGN  051330                9220      9229      9233      9239#
$DTBL   052770                9591      9626#
$ENDAD  051320                 503      1282      1290      9235#     9349
$ENDCT  051250                1274      9222#
$ENDMG  051334                9224      9241#
$ENULL  051351                9227      9244#
$EOP    051214                9212#
$EOPCT  051242                1274*     9219#     9223
$EOT    006576                1657      1659      1661      1703#
$ERFLG  001505                 520#     9252      9282      9284      9290*     9311      9329*     9362
$ERMAX  001517                 526#     1277*     9284      9306*     9311
$ERROR  051636                1268      9328#
$ERRPC  001520                 527#     9339*     9340*     9341      9362     10134     13855     13866     13881     13888     13893     13926     13929
                             13932     13935     13938     13941     13944     13947     13950     13953     13956     13961     13968     13973     13979
                             13996     14013     14015     14016     14018     14021     14024     14026     14030     14035     14067
$ERRTB  001716                 612#    10140
$ERTTL  001514                 524#     9338*     9362
$ESCAP  001700                 584#     1276*     1441*     1492*     9305*     9355      9357      9362
$FILLC  001552                 539#     9452      9483
$FILLS  001551                 538#     9483
$GDADR  001522                 528#
$GDDAT  001526                 530#
```

```
$GET42  051272            9228#
$HD   = 000003              11      12
$HINUM  053106            6816*   6819    6984*   6987    7152*   7155    7317*   7320    7488*   7491    7588*   7599    7648*
                          7651    9649    9657    9662*   9667#
$ICNT   001506             521#   9297*   9298    9300*   9310
$ILLUP  053334            9724    9755#
$ITEMB  001516             525#   3193*   5213*   9341*   9362    9929*  10132
$KTNEX  053560            9795    9821#
$KTOUT  053550            9812    9818#   9850
$KT11   053410            1301*   9793#   9797*   9821*
$LF     001710             588#   9362    9483
$LONUM  053110            6817*   6820    6985*   6988    7153*   7156    7318*   7321    7489*   7492    7589*   7647*   7650
                          9648    9655    9661*   9668#
$LPADR  001510             522#   1278*   9288*   9303*   9308    9310
$LPERR  001512             523#   1279*   1443*   1493*   1542*   1558*   1573*   1599*   1671*   1857*   2044*   2299*   2455*
                          4865*   4897*   4913*   4929*   4945*   4953*   5046*   5064*   5072*   5094*   5148*   5162*   5188*
                          5325*   5375*   5479*   5529*   5629*   5708*   5909*   6236*   6304*   9288    9304*   9310    9354
$LSTAD  053724            9836*   9851#
$LSTBK  053726            1303*   1305    1316    9837*   9852#  10107
$MTMOU  053672            9813    9843#
$MXCNT  051634            9301    9310#
$NULL   001550             537#   9454    9483
$NWTST= 000001            1434#   1436    1482#   1484    1524#   1526    1634#   1636    1704#   1706    1968#   1970    2193#
                          2195    2396#   2398    2588#   2590    2661#   2663    2779#   2781    3071#   3073    3284#   3286
                          3564#   3566    3860#   3862    4156#   4158    4293#   4295    4427#   4429    4588#   4590    4749#
                          4751    4830#   4832    5014#   5016    5130#   5132    5297#   5299    5451#   5453    5606#   5608
                          5844#   5846    6194#   6196    6606#   6608    7676#   7678    7919#   7921    7964#   7966    7994#
                          7996    8019#   8021    8049#   8051    8084#   8086    8202#   8204    8285#   8287    8405#   8407
                          8536#   8538    8570#   8572    8685#   8687    8742#   8744    8859#   8861    8937#   8939    8973#
                          8975    9008#   9010    9043#   9045
$OCNT   052560            9517*   9546*   9559#
$OCTVL  054032            9868    9893#
$OMODE  052562            9512*   9516*   9521    9524*   9535*   9561#
$OVER   051620            9262    9281    9289    9299    9307#
$PASS   001500             518#   1656    9216*   9217*   9225    9241    9295    9311
$PWRAD  053330            9753#
$PWRDN  053206            1272    9724#   9748
$PWRMG  053324            9751#
$PWRUP  053254            9733    9738#
$QUES   001706             586#   9362
$RAND   053010            6818    6986    7154    7319    7490    7598    7649    9644#
$RDCHR= ****** U          9700
$RDDEC= ****** U          9700
$RDLIN= ****** U          9700
$RDOCT= ****** U          9700
$REGAD  001554             541#
$REG0   001556             543#   7930*   7936*   7941*   7947*   7954*   7960*   7976*   7984*   7990*   8007*   8013*   8034*
                          8042*   8064*   8076*   8132*   8146*   8182*   8251*   8262*   8335*   8349*   8385*   8460*   8506*
                          8559*   8626*   8652*   8714*   8729*   8780*   8795*   8802*   8817*   8832*   8840*   8897*   8907*
                          8916*   8951*   8958*   8965*   8987*   8994*   9001*   9022*   9029*   9036*   9120*   9141*   9173*
                         14015   14016   14018   14021   14024   14026
$REG1   001560             544#   8147*   8183*   8252*   8255*   8263*   8350*   8386*   8461*   8464*   8507*   8510*   8627*
                          8630*   8653*   8656*  .8715*   8730*   8818*   8898*   8952*   8959*   8966*   8988*   8995*   9002*
                          9023*   9030*   9037*   9121*   9142*   9174*  14016   14018   14021   14024   14026
$REG10  001576             551#
$REG11  001600             552#
```

```
$REG12   001602        553#
$REG13   001604        554#
$REG14   001606        555#
$REG15   001610        556#
$REG16   001612        557#
$REG17   001614        558#
$REG2    001562        545#    8148*    8184*    8256*    8264*    8351*    8387*    8465*    8511*    8631*    8657*    8716*    8731*
                      8819*    9122*    9143*    9175*   14016    14018    14021    14026
$REG20   001616        559#
$REG21   001620        560#
$REG22   001622        561#
$REG23   001624        562#
$REG3    001564        546#    8820*    9123*    9144*    9176*   14018    14021    14026
$REG4    001566        547#    9124*    9145*    9177*   14018    14026
$REG5    001570        548#
$REG6    001572        549#
$REG7    001574        550#
$RESRE   052064       9397#    9701
$SAVRE   052026       9381#    9700
$SAVR6   053340       9732*    9738     9739*    9740*    9757#
$SCOPE   051354       1266     9260#
$SETUP=  000037        451#    1266     1268     1270     1272     1274     1275     1276     1278     1282     1290     9214     9346
$SIZE    053342       1302     9781#
$SIZEX   053622       9820     9830#
$STUP =  177777        451#
$SVLAD   051572       9270     9302#
$SVPC =  000204        501#     506
$SWR  =  167400         11       12#      13#      21       22       23       24       25       26       27       28      583      584
                       585     1275     1276     1278     1279     1441     1492     1532     1645     1763     1991     2207     2410
                      2598     2670     2802     3082     3322     3589     3885     4169     4302     4440     4597     4773     4838
                      5022     5138     5315     5469     5614     5867     6209     6665     7686     7927     7972     8002     8028
                      8057     8104     8218     8307     8418     8546     8591     8700     8749     8874     8942     8978     9013
                      9048     9209     9215     9230     9240     9241     9252     9253     9254     9255     9256     9261     9273
                      9275     9276     9282     9283     9284     9291     9292     9293     9304     9307     9310     9319     9320
                      9321     9322     9323     9324     9332     9335     9342     9346     9352     9362
$SWRMK=  000200         14#      28       29     9256     9257     9277     9279
$TAB     065076      10217    11370#   13876    13903    13913    13922
$TIMES   001676        583#    1275*    1763*    1991*    2207*    2410*    2598*    2670*    2802*    3082*    3322*    3589*    3885*
                      4169*    4302*    4440*    4597*    4838*    5022*    5138*    5315*    5469*    5867*    6209*    8942*    8978*
                      9013*    9215*    9291*    9298     9301*    9310
$TKB     001542        534#    1429*   10046    10071*
$TKS     001540        533#    1263     1430*   10072*
$TMP0    001626        563#    1768*    1996*    2212*    2415*    2603*    2675*    2806*    3088*    3327*    3596*    3892*    4174*
                      4307*    4445*    4602*    4777*    4843*    5027*    5143*    5320*    5474*    5619*    5872*    6214*    6669*
                      7690*   13855    13858    13866    13876    13881    13885    13888    13891    13893    13897    13907    13917
                     13926    13929    13932    13935    13938    13941    13944    13947    13950    13953    13956    13961    13964
                     13968    13973    13979    13988    13996    14005    14013    14030    14035    14052    14055    14060
$TMP1    001630        564#    1931*    2161*    2360*    2557*    2958*    2978*    3194*    3439*    4887*    4893*    4903*    4909*
                      4919*    4925*    4935*    4941*    5053*    5077*    5084*    5099*    5108*    5171*    5195*    5214*    5258*
                      5286*    5341*    5387*    5495*    5541*    5661*    5823*    5830     5927*    5942*    5956*    5971*    6030*
                      6050*    6069*    6089*    6184*    6255*    6270*    6284*    6299*    6330*    6347*    6366*    6383*    6402*
                      6419*    6438*    6455*    6482*    6505*    6528*    6551*    6573*    6577*    6586*    6591*    6950*    7118*
                      7282*    7451*    7551*    7713*    7753*    7776*    7799*    9898*    9915*    9928*   13855    13858    13876
                     13881    13885    13888    13891    13893    13897    13907    13917    13929    13935    13938    13941    13947
                     13953    13964    14052    14055    14060
$TMP10   001646        571#    3252*    3820*    4118*    4253*    4284*    4388*    4419*    4549*    4580*    4709*    4740*    5900*
```

```
                    5901*   6233*   6234*  13900   13910   13919   13982   13988   13999   14005
$TMP11   001650      572#   4254*   4285*   4389*   4420*   4550*   4581*   4710*   4741*  13982   13999
$TMP12   001652      573#   4255*   4275*   4390*   4410*   4551*   4571*   4711*   4731*  13979   13988   13996   14005
$TMP13   001654      574#   4256*   4391*   4552*   4712*
$TMP14   001656      575#   4250*   4251*   4385*   4386*   4546*   4547*   4706*   4707*  13982   13999
$TMP15   001660      576#   4252*   4387*   4548*   4708*
$TMP16   001662      577#  13979   13996
$TMP17   001664      578#   4242*   4377*   4538*   4698*
$TMP2    001632      565#   2630*   2646*   2738*   2763*   2959*   2979*   3184*   3248*   3440*   3441*   3463*   3481*
                    3699*   3771*   3813*   3998*   4071*   4111*   4277*   4412*   4573*   4733*   4811*   4888*   4889*
                    4894*   4904*   4905*   4910*   4920*   4921*   4926*   4936*   4937*   4942*   4951*   4957*   5054*
                    5078*   5085*   5100*   5109*   5172*   5173*   5196*   5215*   5260*   5340*   5369*   5386*   5415*
                    5494*   5523*   5540*   5569*   5662*   5663*   5824*   5926*   5940*   5941*   5955*   5969*   5970*
                    6029*   6048*   6049*   6068*   6087*   6088*   6183*   6254*   6268*   6269*   6283*   6297*   6298*
                    6329*   6346*   6365*   6382*   6401*   6418*   6437*   6454*   6480*   6481*   6503*   6504*   6526*
                    6527*   6549*   6550*   6580*   6584    6595*   6599    6952*   7120*   7284*   7453*   7552*   7714*
                    7754*   7777*   7800*   9899*   9916*  13855   13858   13866   13876   13885   13893   13897   13907
                   13917   13938   13941   13956   13961   13968   13973   13991   14008   14013   14030   14035   14052
                   14055   14060
$TMP20   001666      579#   4244*   4379*   4540*   4700*
$TMP21   001670      580#
$TMP22   001672      581#
$TMP23   001674      582#
$TMP3    001634      566#   1933*   2163*   2361*   2558*   2631*   2647*   2739*   2764*   2960*   2980*   3247*   3438*
                    3464*   3482*   3700*   3772*   3815*   3999*   4072*   4113*   4245*   4278*   4380*   4413*   4541*
                    4574*   4701*   4734*   5086*   5110*   5216*   5261*   5287*   5370*   5416*   5524*   5570*   5825*
                    6166*   6171*   6175*   6180*   6581*   6596*   6951*   7119*   7283*   7452*   7715*   7755*   7778*
                    7801*   9900*   9917*  13876   13881   13885   13888   13893   13897   13907   13917   13929   13935
                   13947   13953   13956   13968   13973   13979   13996   14030   14035   14052   14060
$TMP4    001636      567#   1934*   2164*   2362*   2559*   2632*   2648*   2740*   2765*   2981*   3250*   3465*   3483*
                    3701*   3773*   3816*   3823*   3828    4000*   4073*   4114*   4121*   4126    4246*   4279*   4280*
                    4381*   4414*   4415*   4542*   4575*   4576*   4702*   4735*   4736*   5087*   5111*   5217*   5263*
                    5288*   5371*   5417*   5525*   5571*   5826*   6165*   6169*   6170*   6174*   6178*   6179*   6574*
                    6578*   6587*   6588*   6592*   6593*   9914*  13858   13876   13881   13885   13888   13893   13897
                   13907   13917   13941   13973   13991   14008
$TMP5    001640      568#   2628*   2645*   2736*   2762*   2982*   3253*   3817*   4115*   4247*   4281*   4382*   4416*
                    4543*   4577*   4703*   4737*   5264*   5265*   5289*   5828*   5896*   6229*  13858   13881   13900
                   13910   13919   13956   13973   13979   13996
$TMP6    001642      569#   3254*   3818*   4116*   4241*   4248*   4282*   4376*   4383*   4417*   4537*   4544*   4578*
                    4697*   4704*   4738*   5266*   5829*   5897*   5898*   6230*   6231*  13858   13876   13900   13910
                   13919   13982   13988   13999   14005
$TMP7    001644      570#   3251*   3819*   4117*   4249*   4283*   4384*   4418*   4545*   4579*   4705*   4739*   5262*
                    5899*   6232*  13876   13900   13910   13919   13973   13991   14008
$TN    = 000061       11#   1434    1441#   1482    1492#   1524    1532#   1634    1645#   1704    1763#   1764    1765
                    1968    1991#   1992    1993    2193    2207#   2208    2209    2396    2410#   2411    2412    2588
                    2598#   2599    2600    2661    2670#   2671    2672    2779    2802#   2803    3071    3082#   3083
                    3084    3284    3322#   3323    3324    3564    3589#   3590    3591    3860    3885#   3886    3887
                    4156    4169#   4170    4171    4293    4302#   4303    4304    4427    4440#   4441    4442    4588
                    4597#   4598    4599    4749    4773#   4774    4830    4838#   4839    5014    5022#   5023    5130
                    5138#   5139    5297    5315#   5316    5317    5451    5469#   5470    5471    5606    5614#   5844
                    5867#   5868    6194    6209#   6210    6606    6665#   7676    7686#   7919    7927#   7964    7972#
                    7989    7994    8002#   8019    8028#   8049    8057#   8084    8104#   8198    8202    8218#   8277
                    8285    8307#   8401    8405    8418#   8529    8536    8546#   8570    8591#   8677    8685    8700#
                    8727    8736    8742    8749#   8853    8859    8874#   8937    8942#   8946    8973    8978#   8982
                    9008    9013#   9017    9043    9048#
$TPB     001546      536#   9472*   9483
```

```
$TPFLG  001553                    540#   9432    9483
$TPS    001544                    535#   9470    9483
$TRAP   053112                   1270    9679#
$TRP   = 000054                  9686#   9696#   9697#   9698#   9699#   9700#   9701#   9702#   9703    9704#   9705#   9706#   9707#
                                 9708#   9709#   9710#   9711#   9712#   9713#   9714    9715#   9716#   9717#   9718#   9719#
$TRPAD  053132                   9683    9694#
$TSTNM  001502                    519#   1259*   1768    1996    2212    2415    2603    2675    2806    3088    3327    3596    3892
                                 4174    4307    4445    4602    4777    4843    5027    5143    5320    5474    5619    5872    6214
                                 6669    7690    9214*   9252    9280    9302*   9307    9311    9331    9362   14067
$TYPBN= ****** U                 9700
$TYPDS  052564                   9576#   9699
$TYPE   052122                   9432#   9686    9695
$TYPEC  052266                   9451    9458    9465    9470#   9471
$TYPEX  052334                   9476    9478    9481#
$TYPOC  052362                   9515#   9696
$TYPON  052376                   9514    9517#   9698
$TYPOS  052336                   9510#   9697
$XTSTR  051362                   9264#
$$GET4= 000001                   9230#   9232#
$$TRP = 000002                   9685#   9696    9697    9698    9699    9700    9701    9702    9704    9705    9706    9707    9708
                                 9709    9710    9711    9712    9713    9715    9716    9717    9718    9719
$OFILL  052561                   9511*   9515*   9525    9560#
.      = 116712                   471#    475     477#    501     502#    504#    506#    515#    589    1264    1278    1279    1295#
                                 1452    1455#   1500    1503#   1582    1585#   1609    1612#   1677    1680#   3028#   4792    4796
                                 5354    5357#   5403    5406#   5508    5511#   5557    5560#   6569#   7821#   8107    8310    9241
                                 9245    9310    9311    9362    9483    9630#   9735    9756    9893#  10381#  11350#  13851#  14069
                                14072#  14073#
```

```
ADDDPA        1#    1827     2269     3046     3373     3648     3721     3947     4021
ADJUST        1#    5353     5402     5507     5556
BYTLT1        1#    4234     4369     4530     4690
BYTLT2        1#    4274     4409     4570     4730
CHAIN         1#    9230
CLRMAC        1#    9940
CLRRFL        1#
CMPDPA        1#    1807     1838     1906     2051     2070     2137     2249     2280     2334     2461     2479     2531     2896     2924
           3146     3222     3386     3530     3659     3732     3958     4032
CNVDPA        1#    1858     2090
CNVMAP        1#    2300     2499
COMMEN        1#     421#
DD            1#    6797     6965     7133     7298     7466
DRIVER        1#    6800     6968     7136     7301
ENDCOM        1#     421#
ERCLR         1#    9359
ERROR        39#    1462     1467     1471     1475     1511     1516     1550     1567     1596     1624     1689     1696     1903     1937
           2134     2167     2330     2365     2528     2562     2633     2649     2741     2766     2961     2983     3185     3255     3442
           3466     3484     3702     3774     3821     4001     4074     4119     4257     4286     4392     4421     4553     4582     4713
           4742     4812     4890     4895     4906     4911     4922     4927     4938     4943     4952     4958     5055     5079     5088
           5101     5112     5174     5197     5267     5291     5342     5372     5388     5418     5496     5526     5542     5572     5664
           5832     5836     5838     5928     5943     5957     5972     6031     6051     6070     6090     6187     6256     6271     6285
           6300     6331     6348     6367     6384     6403     6420     6439     6456     6483     6506     6529     6552     6583     6598
           6953     7121     7285     7454     7553     7718     7722     7724     7757     7780     7803     7870     7888     7906     7931
           7937     7942     7948     7955     7961     7977     7985     7991     8008     8014     8035     8043     8065     8077     8133
           8149     8185     8257     8265     8336     8352     8388     8466     8512     8560     8632     8658     8717     8732     8781
           8796     8803     8822     8833     8841     8899     8908     8917     8953     8960     8967     8989     8996     9003     9024
           9031     9038     9126     9147     9179     9902     9919    10595    10755    10913    11099    11228
ESCAPE        1#     421#
EXTRA         1#     589
HANREG        1#   10573    10733    10891    11077    11206
IIMAC1        1#    5297     5451
ITEMAC        1#    1060
JJM1          1#    4882     4898     4914     4930
KB         7910#    7924     7969     7999     8025     8054     8101     8215     8304     8415     8543     8588     8697     8746     8871
KKM1          1#    5983     5987     5991     5995     6002     6006     6010     6014
KKM2          1#    6019     6038     6058     6077
KMAC1         1#
KMAC5         1#
MMAC1         1#
MMAC2         1#
MMAC3         1#
MMAC4         1#
MMAC5         1#
MMM1          1#    6324     6341     6360     6377     6396     6413     6432     6449
MMM2          1#    6471     6493     6516     6539
MSG        1704#    1706     1967#    1970     2588#    2590     2778#    2781     3070#    3073     3284#    3286     3563#    3566     3859#
           3862     4748#    4751     4830#    4832     5014#    5016     5130#    5132     5297#    5299     5451#    5453     5605#    5608
           5844#    5846     6193#    6196     7675#    7678     7919#    7921     7964#    7966     7994#    7996     8019#    8021     8049#
           8051     8084#    8086     8202#    8204     8285#    8287     8404#    8407     8536#    8538     8569#    8572     8684#    8687
           8742#    8744     8859#    8861     8935#    8939     8973#    8975     9008#    9010     9043#    9045
MSGS       6605#    6608
MSG1       2192#    2195     2398
MSG167     1434#    1436
MSG170     1482#    1484
MSG171     1524#    1526
```

```
MSG172   1634#   1636
MSG2     2660#   2663
MSG201   5296#   5299    5453
MSG3     3563#   3566    3862    4155#   4158    4429
MSG300  12717#  12759   12801   12843
MSG301  12885#  12925   12966   13007
MSG4     4292#   4295    4590
MULT        1#    421#
NEWTST      1#    421#   1434    1482    1524    1634    1704    1968    2193    2396    2588    2661    2779    3071    3284
         3564    3860    4156    4293    4427    4588    4749    4830    5014    5130    5297    5451    5606    5844    6194
         6606    7676    7919    7964    7994    8019    8049    8084    8202    8285    8405    8536    8570    8685    8742
         8859    8937    8973    9008    9043
NMAC1       1#
NMAC2       1#
NMAC3       1#
POP         1#    421#   9402    9617    9663    9742
PUSH        1#    421#   9382    9576    9644    9726
QQM1        1#   5741    5761    5781
SCOPE      40#   1440    1491    1531    1644    1762    1990    2206    2409    2597    2669    2801    3081    3321    3588
         3884    4168    4301    4439    4596    4772    4837    5021    5137    5314    5468    5613    5866    6208    6664
         7685    7926    7971    8001    8027    8056    8103    8217    8306    8417    8545    8590    8699    8748    8873
         8941    8977    9012    9047    9213
SETMAP      1#   4448    4608
SETMM       1#   2217    2418    2689    2814    3121    3330    3622    3920    6670
SETTRA   9686#   9696    9697    9698    9699    9700    9701    9703    9704    9705    9706    9707    9708    9709    9710
         9711    9712    9714    9715    9716    9717    9718
SETUP       1#    421#   1260
SKBREX      1#   5322    5476
SKIP        1#    421#   7989    8198    8277    8401    8529    8677    8727    8736    8853    8946    8982    9017
SLASH       1#    421#
SPACE     421#
SSKAD       1#   1765    1993    2209    2412    2600    2672    2803    3084    3324    3591    3887    4171    4304    4442
         4599    4774    4839    5023    5139    5317    5471    5614    5868    6210    6665    7686
STARS       1#    421#    481     508     596    1296    1434    1439    1482    1490    1524    1530    1536    1552    1569
         1597    1631    1634    1643    1704    1761    1968    1989    2193    2205    2396    2408    2588    2596    2661
         2668    2779    2800    3071    3080    3284    3320    3564    3587    3860    3883    4156    4167    4293    4300
         4427    4438    4588    4595    4749    4771    4830    4836    5014    5020    5130    5136    5297    5313    5451
         5467    5606    5612    5844    5865    6194    6207    6606    6663    7676    7684    7919    7925    7964    7970
         7994    8000    8019    8026    8049    8055    8084    8102    8202    8216    8285    8305    8405    8416    8536
         8544    8570    8589    8685    8698    8742    8747    8859    8872    8937    8940    8973    8976    9008    9011
         9043    9046    9202    9246    9312    9363    9410    9484    9563    9632    9670    9719    9758    9854   10510
        10568
THIT        1#   5335    5381    5489    5535    5921    5935    5950    5964    6024    6043    6063    6082    6249    6263
         6278    6292    6475    6498    6521    6544
TOMAPO      1#   2710    2725    2749
TOPAR3      1#   2943    2963    3167    3684    3757    3983    4057
TRMTRP   9686#
TSTHIT      1#   5919    5932    5947    5961    6247    6260    6275    6289    6473    6496    6519    6542
TYPBIN      1#    421#
TYPDEC      1#    421#   9225
TYPNAM      1#    421#   1280    1288
TYPNUM      1#    421#
TYPOCS      1#    421#   1311    1316
TYPOCT      1#    421#
TYPTXT      1#    421#
UMAC1       1#
```

```
UMAC2         1#   5424    5434    5578    5588
UMAC3         1#   5367    5413    5521    5567
$$CMRE      508#    543     544     545     546     547     548     549     550     551     552     553     554     555     556
            557     558     559     560     561     562
$$CMTM      508#    563     564     565     566     567     568     569     570     571     572     573     574     575     576
            577     578     579     580     581     582
$$ESCA        1#    421#
$$NEWT        1#    421#   1434    1482    1524    1634    1704    1968    2193    2396    2588    2661    2779    3071    3284
           3564    3860    4156    4293    4427    4588    4749    4830    5014    5130    5297    5451    5606    5844    6194
           6606    7676    7919    7964    7994    8019    8049    8084    8202    8285    8405    8536    8570    8685    8742
           8859    8937    8973    9008    9043
$$SET      9686#   9696    9697    9698    9699    9700    9701    9703    9704    9705    9706    9707    9708    9709    9710
           9711    9712    9714    9715    9716    9717    9718
$$SKIP        1#    421#   7989    8198    8277    8401    8529    8677    8727    8736    8853    8946    8982    9017
.EQUAT        1#
.HEADE        1#
.KT11         1#
.SETUP        1#    451
.SWRHI        1#     16
.SWRLO       29#
.$ACT1        1#    481
.$CATC        1#    468
.$CMTA        1#    508
.$DB2D        1#
.$DB2O        1#   9854
.$DIV         1#
.$EOP         1#   9202
.$ERRO        1#   9312
.$ERRT        1#
.$MULT        1#
.$POWE        1#   9719
.$RAND        1#   9632
.$RDDE        1#
.$RDOC        1#
.$READ        1#
.$SAVE        1#   9363
.$SB2D        1#
.$SB2O        1#
.$SCOP        1#   9246
.$SIZE        1#   9758
.$SUPR        1#
.$TRAP        1#   9670
.$TYPB        1#
.$TYPD        1#   9563
.$TYPE        1#   9410
.$TYPO        1#   9484
.1170         1#     31


. ABS.  116712      000


 ERRORS DETECTED:  0

 DSKZ:CEKBDD.BIN,DSKZ:CEKBDD.LST/CRF/SOL=CEKBDD.SML,CEKBDD.P11
 RUN-TIME: 80 120 18 SECONDS
```

```
RUN-TIME RATIO: 322/219=1.4
CORE USED:  38K  (75 PAGES)
```