PDP11/70

CACHE DIAG PART 2

MD-11-DEKBD-C

EP-DEKBD-C-DL-A

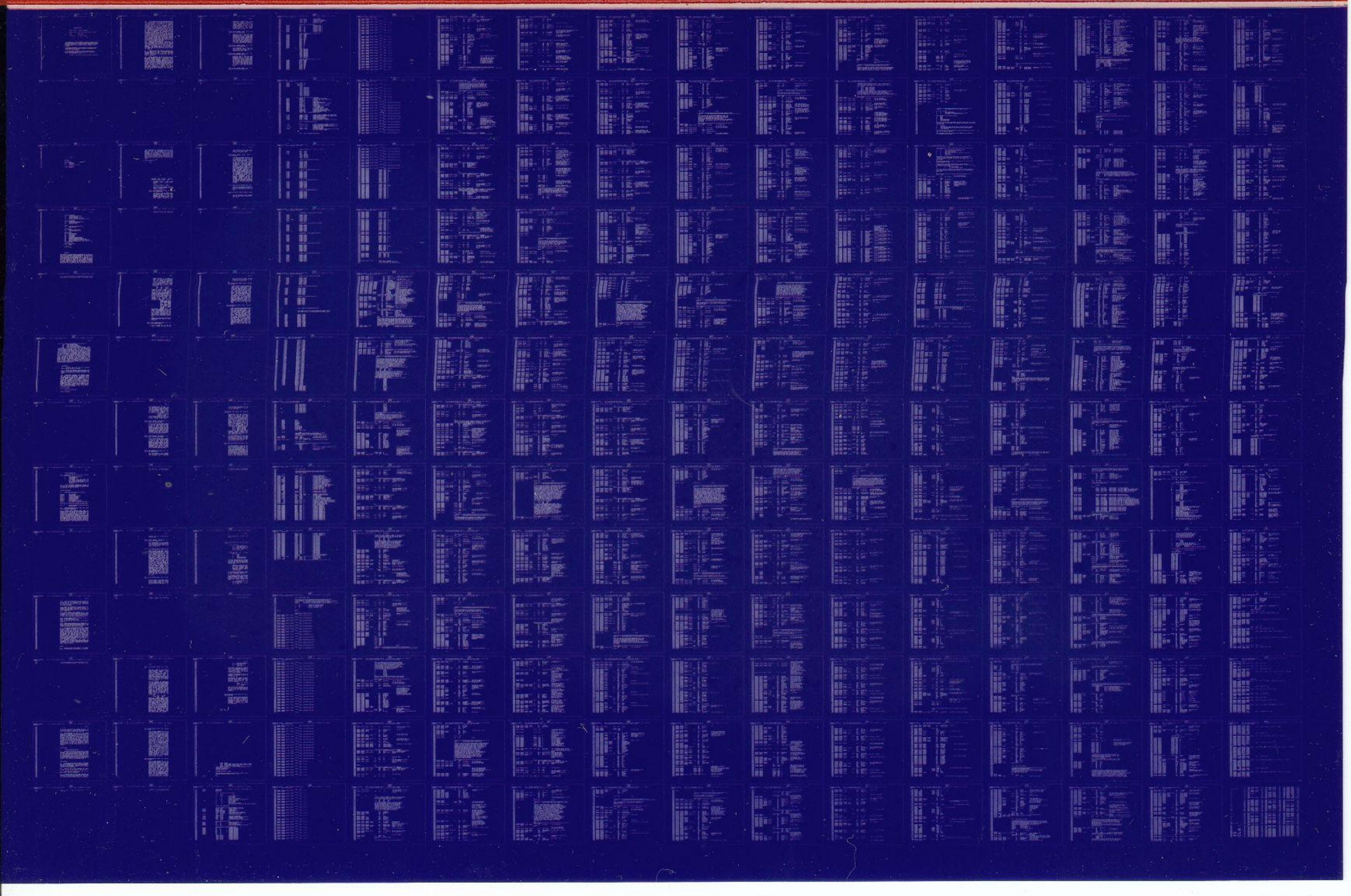
COPYRIGHT © 1976

FICHE 1 OF 2

NOV 1976

digital

MADE IN USA



PDP11/70

CACHE PART 2

MD-11-DEKBD-C

EP-DEKBD-C-DL-A

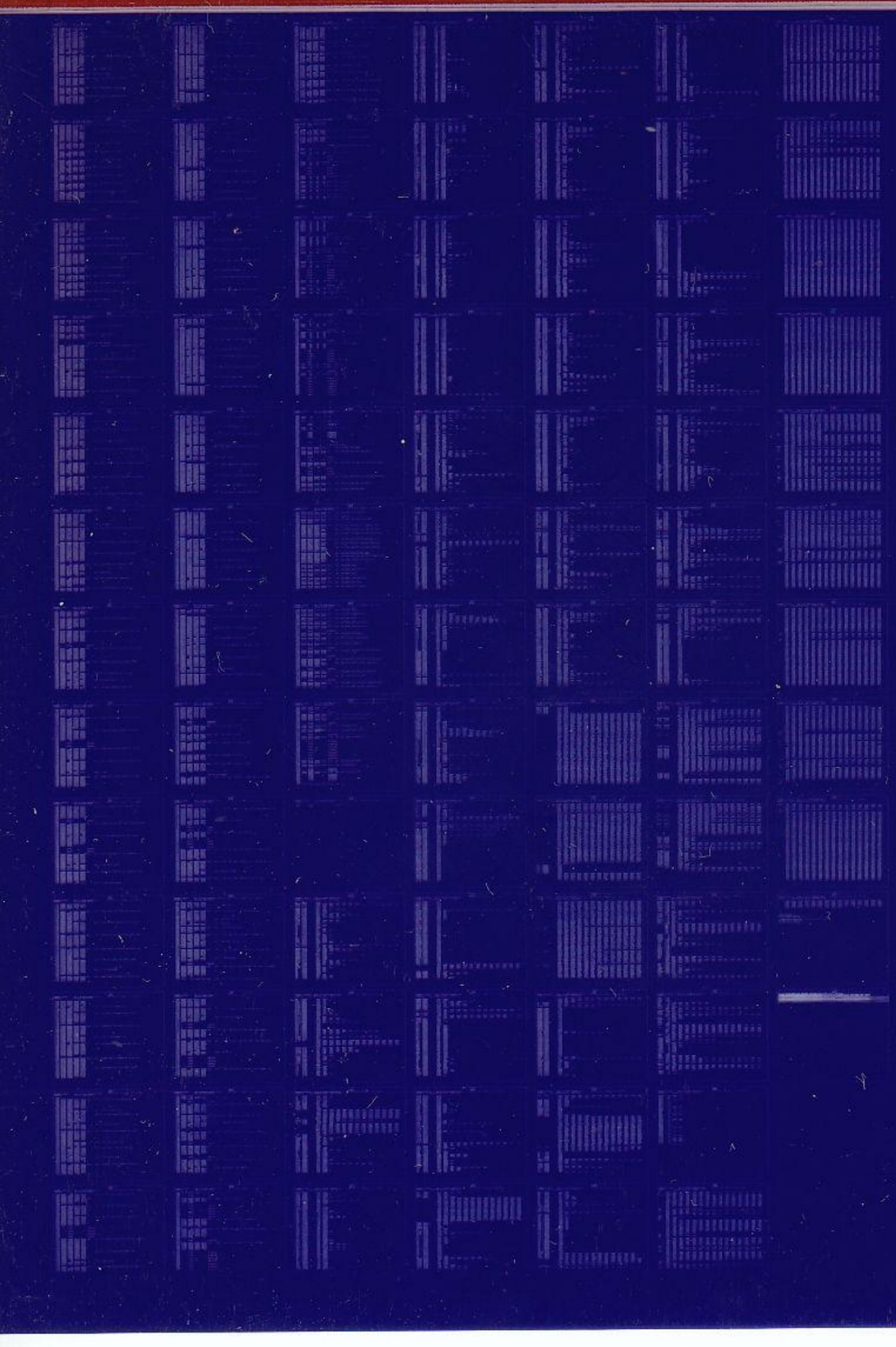
COPYRIGHT © 1976

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA



```
                        .REM !
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DEKBD-C

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

DATE: APRIL,1976

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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT 1975,1976 BY DIGITAL EQUIPMENT CORPORATION

```
    57
```

CONTENTS

1. ABSTRACT

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

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

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

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

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

2.3 PRELIMINARY PROGRAMS

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

3. LOADING PROCEDURE

3.1 METHOD (TO BE SUPPLIED)

4. STARTING PROCEDURE

```
194
195                                    4.1     CONTROL SWITCH SETTINGS (SEE 5.1)
```

4.2 STARTING ADDRESS 200

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

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

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

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS FOR DEKBC:

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

DEKBD USES THE SAME SWITCH SETTINGS AS DEKBC EXCEPT

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

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

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

252 OCCURRED.

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

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

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

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

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

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

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

6. ERRORS

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

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

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

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

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS NONE

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

8. MISCELLANEOUS

8.1 EXECUTION TIME FIRST PASS UNDER 30 SECONDS FOR BOTH PROGRAMS. BUT THIS IS DEVICE DEPENDENT (SEE TEST 31 AND TEST 32). 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 1500.

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

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

```
                        A ONE (1)  MAKING  ITERATIONS  MEANINGLESS  ON  THAT
                        FIRST PASS.
```

8.5 OSCILLOSCOPE SYNC POINTS WHERE EVER POSSIBLE EACH TEST HAS BEEN GIVEN AN OSCILLOSCOPE SYNC POINT (A NOP INSTRUCTION). THE ADDRESS OF THE 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 NOP INSTRUCTION IN CRUCIAL PARTS OF A TEST THE USER WILL HAVE A VERY CONVENIENT SYNC FOR MANY SIGNALS HE MAY WISH TO OBSERVER. THE LIMITATIONS OF THIS PROCEDURE ARE THAT THE USER MUST BE ABLE TO JUDGE (DETERMINE) HOW SOON AFTER THE NOP IN THE PARTICULAR TEST HE IS RUNNING (LOOPING ON) THE SIGNAL HE WISHES TO OBSERVE SHOULD OCCUR. IN MANY CASES THIS WILL BE EASY (E.G. THE ERROR REGISTER TESTS.) BUT IN SOME TESTS THE NOP IS SO FAR FROM THE EXPECTED OCCURRENCE OF THE DESIRED SIGNAL THAT THE PROBLEM BECOMES NONTRIVIAL AND THE EXPERIENCED USER WOULD DO WELL TO FIND OTHER SYNC SIGNALS ORIGINATING IN THE CACHE DEVICE ITSELF TO OBSERVE THE LOGIC.

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

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

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

```
425
426                                 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 DEKBD

COPYRIGHT 1975 DIGITAL EQUIPMENT CORPORATION MAYNARD, MASS. 01754

COPYRIGHT (C) SEPT 11, 1975 DIGITAL EQUIPMENT CORP. MAYNARD, MASS. 01754

PROGRAM BY ANTHONY S. VEZZA

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

TEST 1 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.

TEST 2 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 3 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 4 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 5 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 6 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 7 CACHE ADDRESS MEMORY COMPARATOR TEST

THIS IS A TEST OF THE CACHE ADDRESS MEMORY ADDRESS COMPARATORS. THIS IS A CIRCUIT MADE UP OF SIX 74S85 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 10 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

```
   657                                  ATTEMPT  TO  MAKE  EVERY  ADDRESS IN
   658                                  AVAILABLE  MEMORY  A  HIT  IN  EACH
```

GROUP.

TEST 11 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.

PAGE 13

TEST 12 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 13 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 14 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 15 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 16 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 17 CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ZEROES TEST

```
829
830                                     THIS IS ANOTHER  TEST  OF  THE  BYTE
```

MASK GENERATION LIGIC. HERE UNIBUS DATOB'S WILL MOVE ZEROES INTO A BACKROUND PATTERN OF ONES.

TEST 20 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 21 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 22 CACHE DATA MEMORY ADDRESS DRIVERS TEST

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

```
887
888                                    TEST 23 CACHE DATA MEMORY COUNT PATTERN TEST
```

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

TEST 24 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 25 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.

```
945                                             TEST 26 CACHE DATA MEMORY WORST  CASE  NOISE
946                                             TEST
```

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

TEST 27 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 30 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

```
1003                                    IS  WRITTEN.  AFTER ALL THE PATTERNS
1004                                    HAVE BEEN CHECKED  AND  COMPLEMENTED
```

THE COMPLIMENTED PATTERNS ARE CHECKED.

TEST 31 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

1061 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.

TEST 32 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.

```
]

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

```
                                .ENABL  ABS,AMA
                                .MCALL  .HEADER,.SWRHI,.1170,.SETUP,.$CATCH,.$ACT11,.$CMTAG
                                .MCALL  .$EOP,.$SCOPE,.$ERROR,.$SAVE,.$TYPE,.$TYPOCT,.$RAND
                                .MCALL  .$TYPDEC,.$TRAP,.$POWER,.$DB20
                        .TITLE  MAINDEC-11-DEKBD-C      PDP 11/70 CACHE DIAGNOSTIC PART 2
                        ;*COPYRIGHT (C) SEPT 11, 1975
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ANTHONY S. VEZZA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-CO),MAR 21, 1976.
                        ;*
000001                  $TN=1
```

```
                                $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
                                $SWR=167400
                                $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                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
177546                          LKS=    177546          ;;LINE CLOCK (KW11-L) STATUS REGISTER

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

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
                                .EQUIV  R0,R10          ;;GENERAL REGISTER
                                .EQUIV  R1,R11          ;;GENERAL REGISTER
                                .EQUIV  R2,R12          ;;GENERAL REGISTER
                                .EQUIV  R3,R13          ;;GENERAL REGISTER
                                .EQUIV  R4,R14          ;;GENERAL REGISTER
                                .EQUIV  R5,R15          ;;GENERAL REGISTER
```

```
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  SP,KSP          ;;KERNEL STACK POINTER
                                .EQUIV  SP,SSP          ;;SUPERVISOR STACK POINTER
                                .EQUIV  SP,USP          ;;USER STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

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

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

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                          BIT15=  100000
040000                          BIT14=  40000
020000                          BIT13=  20000
010000                          BIT12=  10000
004000                          BIT11=  4000
002000                          BIT10=  2000
001000                          BIT09=  1000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
```

```
                    BIT04=  20
                    BIT03=  10
                    BIT02=  4
                    BIT01=  2
                    BIT00=  1
                    .EQUIV  BIT09,BIT9
                    .EQUIV  BIT08,BIT8
                    .EQUIV  BIT07,BIT7
                    .EQUIV  BIT06,BIT6
                    .EQUIV  BIT05,BIT5
                    .EQUIV  BIT04,BIT4
                    .EQUIV  BIT03,BIT3
                    .EQUIV  BIT02,BIT2
                    .EQUIV  BIT01,BIT1
                    .EQUIV  BIT00,BIT0

                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004              ERRVEC= 4                   ;;TIME OUT AND OTHER ERRORS
000010              RESVEC= 10                  ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014              TBITVEC=14                  ;;"T" BIT
000014              TRTVEC= 14                  ;;TRACE TRAP
000014              BPTVEC= 14                  ;;BREAKPOINT TRAP (BPT)
000020              IOTVEC= 20                  ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024              PWRVEC= 24                  ;;POWER FAIL
000030              EMTVEC= 30                  ;;EMULATOR TRAP (EMT) **ERROR**
000034              TRAPVEC=34                  ;;"TRAP" TRAP
000060              TKVEC=  60                  ;;TTY KEYBOARD VECTOR
000064              TPVEC=  64                  ;;TTY PRINTER VECTOR
000100              LKVEC=  100                 ;;LINE CLOCK (KW11-L) VECTER
000114              CACHVEC=114                 ;;CACHE ERROR INTERRUPT VECTOR
000240              PIRQVEC=240                 ;;PROGRAM INTERRUPT REQUEST VECTOR
000250              MMVEC=  250                 ;;MEMORY MANAGEMENT VECTOR
                    .SBTTL  CACHE   REGISTER DEFINITIONS


1.7740              LOADRS = 177740             ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
177742              HIADRS = 177742             ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
177744              MEMERR = 177744             ;;CACHE ERROR REGISTER
177746              CONTRL = 177746             ;;MEMORY CONTROL REGISTER
177750              MAINT  = 177750             ;;MEMORY MAINTENENCE REGISTER
177752              HITMIS = 177752             ;;HIT MISS REGISTER "1" IMPLIES HIT IN CACHE

                    .SBTTL  CPU REGISTER DEFINITIONS


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


                    .SBTTL  MEMORY MANAGEMENT DEFINITIONS
```

```
1342
1343
1344                                    ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES
1345
1346            177572                  MMR0=   177572
1347            177574                  MMR1=   177574
1348            177576                  MMR2=   177576
1349            172516                  MMR3=   172516
1350                                    .EQUIV  MMR0,SR0
1351                                    .EQUIV  MMR1,SR1
1352                                    .EQUIV  MMR2,SR2
1353                                    .EQUIV  MMR3,SR3
1354
1355                                    ;*USER "I" PAGE DESCRIPTOR REGISTERS
1356
1357            177600                  UIPDR0= 177600
1358            177602                  UIPDR1= 177602
1359            177604                  UIPDR2= 177604
1360            177606                  UIPDR3= 177606
1361            177610                  UIPDR4= 177610
1362            177612                  UIPDR5= 177612
1363            177614                  UIPDR6= 177614
1364            177616                  UIPDR7= 177616
1365
1366                                    ;*USER "D" PAGE DESCRIPTOR REGISTORS
1367
1368            177620                  UDPDR0= 177620
1369            177622                  UDPDR1= 177622
1370            177624                  UDPDR2= 177624
1371            177626                  UDPDR3= 177626
1372            177630                  UDPDR4= 177630
1373            177632                  UDPDR5= 177632
1374            177634                  UDPDR6= 177634
1375            177636                  UDPDR7= 177636
1376
1377                                    ;*USER "I" PAGE ADDRESS REGISTERS
1378
1379            177640                  UIPAR0= 177640
1380            177642                  UIPAR1= 177642
1381            177644                  UIPAR2= 177644
1382            177646                  UIPAR3= 177646
1383            177650                  UIPAR4= 177650
1384            177652                  UIPAR5= 177652
1385            177654                  UIPAR6= 177654
1386            177656                  UIPAR7= 177656
1387
1388                                    ;*USER "D" PAGE ADDRESS REGISTERS
1389
1390            177660                  UDPAR0= 177660
1391            177662                  UDPAR1= 177662
1392            177664                  UDPAR2= 177664
1393            177666                  UDPAR3= 177666
1394            177670                  UDPAR4= 177670
1395            177672                  UDPAR5= 177672
1396            177674                  UDPAR6= 177674
1397            177676                  UDPAR7= 177676
```

```
1398
1399                                    ;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
1400
1401            172200                  SIPDR0= 172200
1402            172202                  SIPDR1= 172202
1403            172204                  SIPDR2= 172204
1404            172206                  SIPDR3= 172206
1405            172210                  SIPDR4= 172210
1406            172212                  SIPDR5= 172212
1407            172214                  SIPDR6= 172214
1408            172216                  SIPDR7= 172216
1409
1410                                    ;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
1411
1412            172220                  SDPDR0= 172220
1413            172222                  SDPDR1= 172222
1414            172224                  SDPDR2= 172224
1415            172226                  SDPDR3= 172226
1416            172230                  SDPDR4= 172230
1417            172232                  SDPDR5= 172232
1418            172234                  SDPDR6= 172234
1419            172236                  SDPDR7= 172236
1420
1421                                    ;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
1422
1423            172240                  SIPAR0= 172240
1424            172242                  SIPAR1= 172242
1425            172244                  SIPAR2= 172244
1426            172246                  SIPAR3= 172246
1427            172250                  SIPAR4= 172250
1428            172252                  SIPAR5= 172252
1429            172254                  SIPAR6= 172254
1430            172256                  SIPAR7= 172256
1431
1432                                    ;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
1433
1434            172260                  SDPAR0= 172260
1435            172262                  SDPAR1= 172262
1436            172264                  SDPAR2= 172264
1437            172266                  SDPAR3= 172266
1438            172270                  SDPAR4= 172270
1439            172272                  SDPAR5= 172272
1440            172274                  SDPAR6= 172274
1441            172276                  SDPAR7= 172276
1442
1443                                    ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
1444
1445            172300                  KIPDR0= 172300
1446            172302                  KIPDR1= 172302
1447            172304                  KIPDR2= 172304
1448            172306                  KIPDR3= 172306
1449            172310                  KIPDR4= 172310
1450            172312                  KIPDR5= 172312
1451            172314                  KIPDR6= 172314
1452            172316                  KIPDR7= 172316
1453
```

```
1454                                    ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
1455
1456            172320                  KDPDR0= 172320
1457            172322                  KDPDR1= 172322
1458            172324                  KDPDR2= 172324
1459            172326                  KDPDR3= 172326
1460            172330                  KDPDR4= 172330
1461            172332                  KDPDR5= 172332
1462            172334                  KDPDR6= 172334
1463            172336                  KDPDR7= 172336
1464
1465                                    ;*KERNEL "I" PAGE ADDRESS REGISTERS
1466
1467            172340                  KIPAR0= 172340
1468            172342                  KIPAR1= 172342
1469            172344                  KIPAR2= 172344
1470            172346                  KIPAR3= 172346
1471            172350                  KIPAR4= 172350
1472            172352                  KIPAR5= 172352
1473            172354                  KIPAR6= 172354
1474            172356                  KIPAR7= 172356
1475
1476                                    ;*KERNEL "D" PAGE ADDRESS REGISTERS
1477
1478            172360                  KDPAR0= 172360
1479            172362                  KDPAR1= 172362
1480            172364                  KDPAR2= 172364
1481            172366                  KDPAR3= 172366
1482            172370                  KDPAR4= 172370
1483            172372                  KDPAR5= 172372
1484            172374                  KDPAR6= 172374
1485            172376                  KDPAR7= 172376
1486
1487
1488
1489                                    .SBTTL  UNIBUS MAP REGISTER DEFINITIONS
1490
1491
1492                                    ;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
1493                                    ;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'
1494
1495
1496            170200                  MAPL00 = 170200
1497            170202                  MAPH00 = 170202
1498            170204                  MAPL01 = 170204
1499            170206                  MAPH01 = 170206
1500            170210                  MAPL02 = 170210
1501            170212                  MAPH02 = 170212
1502            170214                  MAPL03 = 170214
1503            170216                  MAPH03 = 170216
1504            170220                  MAPL04 = 170220
1505            170222                  MAPH04 = 170222
1506            170224                  MAPL05 = 170224
1507            170226                  MAPH05 = 170226
1508            170230                  MAPL06 = 170230
1509            170232                  MAPH06 = 170232
```

```
1510            170234                  MAPL07 = 170234
1511            170236                  MAPH07 = 170236
1512            170240                  MAPL10 = 170240
1513            170242                  MAPH10 = 170242
1514            170244                  MAPL11 = 170244
1515            170246                  MAPH11 = 170246
1516            170250                  MAPL12 = 170250
1517            170252                  MAPH12 = 170252
1518            170254                  MAPL13 = 170254
1519            170256                  MAPH13 = 170256
1520            170260                  MAPL14 = 170260
1521            170262                  MAPH14 = 170262
1522            170264                  MAPL15 = 170264
1523            170266                  MAPH15 = 170266
1524            170270                  MAPL16 = 170270
1525            170272                  MAPH16 = 170272
1526            170274                  MAPL17 = 170274
1527            170276                  MAPH17 = 170276
1528            170300                  MAPL20 = 170300
1529            170302                  MAPH20 = 170302
1530            170304                  MAPL21 = 170304
1531            170306                  MAPH21 = 170306
1532            170310                  MAPL22 = 170310
1533            170312                  MAPH22 = 170312
1534            170314                  MAPL23 = 170314
1535            170316                  MAPH23 = 170316
1536            170320                  MAPL24 = 170320
1537            170320                  MAPH24 = 170320
1538            170324                  MAPL25 = 170324
1539            170326                  MAPH25 = 170326
1540            170330                  MAPL26 = 170330
1541            170332                  MAPH26 = 170332
1542            170334                  MAPL27 = 170334
1543            170336                  MAPH27 = 170336
1544            170340                  MAPL30 = 170340
1545            170342                  MAPH30 = 170342
1546            170344                  MAPL31 = 170344
1547            170346                  MAPH31 = 170346
1548            170350                  MAPL32 = 170350
1549            170352                  MAPH32 = 170352
1550            170354                  MAPL33 = 170354
1551            170356                  MAPH33 = 170356
1552            170360                  MAPL34 = 170360
1553            170362                  MAPH34 = 170362
1554            170364                  MAPL35 = 170364
1555            170366                  MAPH35 = 170366
1556            170370                  MAPL36 = 170370
1557            170372                  MAPH36 = 170372
1558            170374                  MAPL37 = 170374
1559            170376                  MAPH37 = 170376
1560                                    .EQUIV  MAPL00,MAPL0
1561                                    .EQUIV  MAPH00,MAPH0
1562                                    .EQUIV  MAPL01,MAPL1
1563                                    .EQUIV  MAPH01,MAPH1
1564                                    .EQUIV  MAPL02,MAPL2
1565                                    .EQUIV  MAPH02,MAPH2
```

```
                                        .EQUIV  MAPLO3,MAPL3
                                        .EQUIV  MAPHO3,MAPH3
                                        .EQUIV  MAPLO4,MAPL4
                                        .EQUIV  MAPHO4,MAPH4
                                        .EQUIV  MAPLO5,MAPL5
                                        .EQUIV  MAPHO5,MAPH5
                                        .EQUIV  MAPLO6,MAPL6
                                        .EQUIV  MAPHO6,MAPH6
                                        .EQUIV  MAPLO7,MAPL7
                                        .EQUIV  MAPHO7,MAPH7


                    000011              TAB=11
                    000044              S1MO=44
                    000030              SOM1=30
                    000054              S1MOM1=54
                    000034              SOMOM1=34
                    000014              M1MO=14
                    000014              MOM1=M1MO
                    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
                    000174              .=174
000174  000000     DISPREG: .WORD  0                ;;SOFTWARE DISPLAY REGISTER
000176  000000     SWREG:   .WORD  0                ;;SOFTWARE SWITCH REGISTER
                    .SBTTL  STARTING ADDRESS(ES)
000200  000137  003752      JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

                    .SBTTL  ACT11 HOOKS

                    ;;******************************************************************
                    ;HOOKS REQUIRED BY ACT11
                    000204          $SVPC=.                 ;SAVE PC
                    000046          .=46
000046  041300      $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                    000052          .=52
000052  000000      .WORD   0               ;;2)SET LOC.52 TO ZERO
                    000204          .=$SVPC                 ;; RESTORE PC
```

```
                                .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             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001503  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001504  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001506  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001510  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001512  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001514  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001515  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001516  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001520  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001522  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001524  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001526  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001530  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001532  000000                  .WORD   0
001534  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001535  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001536  000000                  .WORD   0
001540  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001542  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001544  177560          $TKS:   177560                  ;;TTY KBD STATUS
001546  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001550  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001552  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001554  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001555  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001556  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001557  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001560  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001562  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001564  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001566  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001570  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001572  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001574  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001576  000000          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001600  000000          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001602  000000          $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001604  000000          $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
001606  000000          $REG12: .WORD   0               ;;CONTAINS (($REGAD)+24)
001610  000000          $REG13: .WORD   0               ;;CONTAINS (($REGAD)+26)
001612  000000          $REG14: .WORD   0               ;;CONTAINS (($REGAD)+30)
001614  000000          $REG15: .WORD   0               ;;CONTAINS (($REGAD)+32)
001616  000000          $REG16: .WORD   0               ;;CONTAINS (($REGAD)+34)
001620  000000          $REG17: .WORD   0               ;;CONTAINS (($REGAD)+36)
001622  000000          $REG20: .WORD   0               ;;CONTAINS (($REGAD)+40)
```

```
1673  001624  000000                $REG21: .WORD   0               ;;CONTAINS (($REGAD)+42)
1674  001626  000000                $REG22: .WORD   0               ;;CONTAINS (($REGAD)+44)
1675  001630  000000                $REG23: .WORD   0               ;;CONTAINS (($REGAD)+46)
1676  001632  000000                $TMP0:  .WORD   0               ;;USER DEFINED
1677  001634  000000                $TMP1:  .WORD   0               ;;USER DEFINED
1678  001636  000000                $TMP2:  .WORD   0               ;;USER DEFINED
1679  001640  000000                $TMP3:  .WORD   0               ;;USER DEFINED
1680  001642  000000                $TMP4:  .WORD   0               ;;USER DEFINED
1681  001644  000000                $TMP5:  .WORD   0               ;;USER DEFINED
1682  001646  000000                $TMP6:  .WORD   0               ;;USER DEFINED
1683  001650  000000                $TMP7:  .WORD   0               ;;USER DEFINED
1684  001652  000000                $TMP10: .WORD   0               ;;USER DEFINED
1685  001654  000000                $TMP11: .WORD   0               ;;USER DEFINED
1686  001656  000000                $TMP12: .WORD   0               ;;USER DEFINED
1687  001660  000000                $TMP13: .WORD   0               ;;USER DEFINED
1688  001662  000000                $TMP14: .WORD   0               ;;USER DEFINED
1689  001664  000000                $TMP15: .WORD   0               ;;USER DEFINED
1690  001666  000000                $TMP16: .WORD   0               ;;USER DEFINED
1691  001670  000000                $TMP17: .WORD   0               ;;USER DEFINED
1692  001672  000000                $TMP20: .WORD   0               ;;USER DEFINED
1693  001674  000000                $TMP21: .WORD   0               ;;USER DEFINED
1694  001676  000000                $TMP22: .WORD   0               ;;USER DEFINED
1695  001700  000000                $TMP23: .WORD   0               ;;USER DEFINED
1696  001702  000000                $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
1697  001704  000000                $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
1698  001706  177607  000377        $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
1699  001712     077                $QUES:  .ASCII  /?/             ;;QUESTION MARK
1700  001713     015                $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
1701  001714  000012                $LF:    .ASCIZ  <12>            ;;LINE FEED
1702                                ;;**************************************************************
```

```
                                        .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  055371  067211  072310                  .WORD   EM1,DH1,DT1,DF1
001724  072011
                                        ;ITEM 2
001726  055456  067264  072322                  .WORD   EM2,DH2,DT2,DF2
001734  072015
                                        ;ITEM 3
001736  055650  067264  072322                  .WORD   EM3,DH3,DT3,DF3
001744  072015
                                        ;ITEM 4
001746  055764  067264  072322                  .WORD   EM4,DH4,DT4,DF4
001754  072015
                                        ;ITEM 5
001756  056077  067365  072340                  .WORD   EM5,DH5,DT5,DF5
001764  072023
                                        ;ITEM 6
001766  056157  067365  072340                  .WORD   EM6,DH6,DT6,DF6
001774  072023
                                        ;ITEM 7
001776  056237  067365  072340                  .WORD   EM7,DH7,DT7,DF7
002004  072023
                                        ;ITEM 10
002006  056331  067365  072340                  .WORD   EM10,DH10,DT10,DF10
002014  072023
                                        ;ITEM 11
002016  056422  067417  072362                  .WORD   EM11,DH11,DT11,DF11
002024  072033
                                        ;ITEM 12
002026  056510  067473  072406                  .WORD   EM12,DH12,DT12,DF12
002034  072044
                                        ;ITEM 13
002036  056635  067566  072424                  .WORD   EM13,DH13,DT13,DF13
002044  072052
                                        ;ITEM 14
002046  056715  067641  072440                  .WORD   EM14,DH14,DT14,DF14
002054  072057
                                        ;ITEM 15
002056  056754  067734  072454                  .WORD   EM15,DH15,DT15,DF15
```

```
1759  002064  072064
1760                                  ;ITEM 16
1761  002066  057024  067760  072462          .WORD   EM16,DH16,DT16,DF16
1762  002074  072066
1763                                  ;ITEM 17
1764  002076  057100  070046  072500          .WORD   EM17,DH17,DT17,DF17
1765  002104  072074
1766                                  ;ITEM 20
1767  002106  057100  070046  072562          .WORD   EM20,DH20,DT20,DF20
1768  002114  072074
1769                                  ;ITEM 21
1770  002116  057161  070126  072644          .WORD   EM21,DH21,DT21,DF21
1771  002124  072124
1772                                  ;ITEM 22
1773  002126  057245  070201  072722          .WORD   EM22,DH22,DT22,DF22
1774  002134  072152
1775                                  ;ITEM 23
1776  002136  057461  070250  072732          .WORD   EM23,DH23,DT23,DF23
1777  002144  072155
1778                                  ;ITEM 24
1779  002146  057245  070201  072746          .WORD   EM24,DH24,DT24,DF24
1780  002154  072152
1781                                  ;ITEM 25
1782  002156  057461  070250  072756          .WORD   EM25,DH25,DT25,DF25
1783  002164  072155
1784                                  ;ITEM 26
1785  002166  057615  070340  072772          .WORD   EM26,DH26,DT26,DF26
1786  002174  072162
1787                                  ;ITEM 27
1788  002176  057762  070405  073004          .WORD   EM27,DH27,DT27,DF27
1789  002204  072166
1790                                  ;ITEM 30
1791  002206  057245  070201  073020          .WORD   EM30,DH30,DT30,DF30
1792  002214  072152
1793                                  ;ITEM 31
1794  002216  060127  070250  073030          .WORD   EM31,DH31,DT31,DF31
1795  002224  072155
1796                                  ;ITEM 32
1797  002226  057245  070201  073044          .WORD   EM32,DH32,DT32,DF32
1798  002234  072152
1799                                  ;ITEM 33
1800  002236  060127  070250  073054          .WORD   EM33,DH33,DT33.DF33
1801  002244  072155
1802                                  ;ITEM 34
1803  002246  060266  070477  073070          .WORD   EM34,DH34,DT34,DF34
1804  002254  072173
1805                                  ;ITEM 35
1806  002256  060372  070477  073070          .WORD   EM35,DH35,DT35,DF35
1807  002264  072173
1808                                  ;ITEM 36
1809  002266  060501  070557  073104          .WORD   EM36,DH36,DT36,DF36
1810  002274  072200
1811                                  ;ITEM 37
1812  002276  060633  070624  073116          .WORD   EM37,DH37,DT37,DF37
1813  002304  072204
1814                                  ;ITEM 40
```

```
1815  002306  060715  070756  073144          .WORD   EM40,DH40,DT40,DF40
1816  002314  072216
1817                                  ;ITEM 41
1818  002316  061070  070711  073132          .WORD   EM41,DH41,DT41,DF41
1819  002324  072212
1820                                  ;ITEM 42
1821  002326  061254  070711  073132          .WORD   EM42,DH42,DT42,DF42
1822  002334  072212
1823                                  ;ITEM 43
1824  002336  061444  070756  073144          .WORD   EM43,DH43,DT43,DF43
1825  002344  072216
1826                                  ;ITEM 44
1827  002346  061572  071025  073166          .WORD   EM44,DH44,DT44,DF44
1828  002354  072226
1829                                  ;ITEM 45
1830  002356  061766  071025  073166          .WORD   EM45,DH45,DT45,DF45
1831  002364  072226
1832                                  ;ITEM 46
1833  002366  062165  071120  073226          .WORD   EM46,DH46,DT46,DF46
1834  002374  072245
1835                                  ;ITEM 47
1836  002376  062306  071120  073226          .WORD   EM47,DH47,DT47,DF47
1837  002404  072245
1838                                  ;ITEM 50
1839  002406  061572  071025  073260          .WORD   EM50,DH50,DT50,DF50
1840  002414  072226
1841                                  ;ITEM 51
1842  002416  061766  071025  073260          .WORD   EM51,DH51,DT51,DF51
1843  002424  072226
1844                                  ;ITEM 52
1845  002426  062165  071120  073320          .WORD   EM52,DH52,DT52,DF52
1846  002434  072245
1847                                  ;ITEM 53
1848  002436  062306  071120  073320          .WORD   EM53,DH53,DT53,DF53
1849  002444  072245
1850                                  ;ITEM 54
1851  002446  062432  071144  073352          .WORD   EM54,DH54,DT54,DF54
1852  002454  072261
1853                                  ;ITEM 0
1854  002456  000000  000000  000000          .WORD   0,0,0,0
1855  002464  000000
1856                                  ;ITEM 0
1857  002466  000000  000000  000000          .WORD   0,0,0,0
1858  002474  000000
1859                                  ;ITEM 0
1860  002476  000000  000000  000000          .WORD   0,0,0,0
1861  002504  000000
1862                                  ;ITEM 0
1863  002506  000000  000000  000000          .WORD   0,0,0,0
1864  002514  000000
1865                                  ;ITEM 0
1866  002516  000000  000000  000000          .WORD   0,0,0,0
1867  002524  000000
1868                                  ;ITEM 0
1869  002526  000000  000000  000000          .WORD   0,0,0,0
1870  002534  000000
```

```
1871                                            ;ITEM 0
1872  002536  000000  000000  000000                    .WORD   0,0,0,0
1873  002544  000000
1874                                            ;ITEM 0
1875  002546  000000  000000  000000                    .WORD   0,0,0,0
1876  002554  000000
1877                                            ;ITEM 0
1878  002556  000000  000000  000000                    .WORD   0,0,0,0
1879  002564  000000
1880                                            ;ITEM 0
1881  002566  000000  000000  000000                    .WORD   0,0,0,0
1882  002574  000000
1883                                            ;ITEM 0
1884  002576  000000  000000  000000                    .WORD   0,0,0,0
1885  002604  000000
1886                                            ;ITEM 0
1887  002606  000000  000000  000000                    .WORD   0,0,0,0
1888  002614  000000
1889                                            ;ITEM 0
1890  002616  000000  000000  000000                    .WORD   0,0,0,0
1891  002624  000000
1892                                            ;ITEM 0
1893  002626  000000  000000  000000                    .WORD   0,0,0,0
1894  002634  000000
1895                                            ;ITEM 0
1896  002636  000000  000000  000000                    .WORD   0,0,0,0
1897  002644  000000
1898                                            ;ITEM 0
1899  002646  000000  000000  000000                    .WORD   0,0,0,0
1900  002654  000000
1901
1902                                            ;ITEM 0
1903  002656  000000  000000  000000                    .WORD   0,0,0,0
1904  002664  000000
1905                                            ;ITEM 0
1906  002666  000000  000000  000000                    .WORD   0,0,0,0
1907  002674  000000
1908                                            ;ITEM 0
1909  002676  000000  000000  000000                    .WORD   0,0,0,0
1910  002704  000000
1911                                            ;ITEM 0
1912  002706  000000  000000  000000                    .WORD   0,0,0,0
1913  002714  000000
1914                                            ;ITEM 0
1915  002716  000000  000000  000000                    .WORD   0,0,0,0
1916  002724  000000
1917                                            ;ITEM 0
1918  002726  000000  000000  000000                    .WORD   0,0,0,0
1919  002734  000000
1920                                            ;ITEM 0
1921  002736  000000  000000  000000                    .WORD   0,0,0,0
1922  002744  000000
1923                                            ;ITEM 0
1924  002746  000000  000000  000000                    .WORD   0,0,0,0
1925  002754  000000
1926                                            ;ITEM 0
```

```
1927  002756  000000  000000  000000          .WORD   0,0,0,0
1928  002764  000000
1929                                  ;ITEM 0
1930  002766  000000  000000  000000          .WORD   0,0,0,0
1931  002774  000000
1932                                  ;ITEM 0
1933  002776  000000  000000  000000          .WORD   0,0,0,0
1934  003004  000000
1935                                  ;ITEM 0
1936  003006  000000  000000  000000          .WORD   0,0,0,0
1937  003014  000000
1938                                  ;ITEM 0
1939  003016  000000  000000  000000          .WORD   0,0,0,0
1940  003024  000000
1941                                  ;ITEM 0
1942  003026  000000  000000  000000          .WORD   0,0,0,0
1943  003034  000000
1944
1945                                  ;ITEM 0
1946  003036  000000  000000  000000          .WORD   0,0,0,0
1947  003044  000000
1948                                  ;ITEM 0
1949  003046  000000  000000  000000          .WORD   0,0,0,0
1950  003054  000000
1951                                  ;ITEM 0
1952  003056  000000  000000  000000          .WORD   0,0,0,0
1953  003064  000000
1954                                  ;ITEM 0
1955  003066  000000  000000  000000          .WORD   0,0,0,0
1956  003074  000000
1957                                  ;ITEM 0
1958  003076  000000  000000  000000          .WORD   0,0,0,0
1959  003104  000000
1960                                  ;ITEM 0
1961  003106  000000  000000  000000          .WORD   0,0,0,0
1962  003114  000000
1963                                  ;ITEM 0
1964  003116  000000  000000  000000          .WORD   0,0,0,0
1965  003124  000000
1966                                  ;ITEM 0
1967  003126  000000  000000  000000          .WORD   0,0,0,0
1968  003134  000000
1969                                  ;ITEM 0
1970  003136  000000  000000  000000          .WORD   0,0,0,0
1971  003144  000000
1972                                  ;ITEM 0
1973  003146  000000  000000  000000          .WORD   0,0,0,0
1974  003154  000000
1975                                  ;ITEM 0
1976  003156  000000  000000  000000          .WORD   0,0,0,0
1977  003164  000000
1978                                  ;ITEM 0
1979  003166  000000  000000  000000          .WORD   0,0,0,0
1980  003174  000000
1981                                  ;ITEM 0
1982  003176  000000  000000  000000          .WORD   0,0,0,0
```

```
1983  003204  000000
1984                                    ;ITEM 0
1985  003206  000000  000000  000000            .WORD   0,0,0,0
1986  003214  000000
1987
1988                                    ;ITEM 0
1989  003216  000000  000000  000000            .WORD   0,0,0,0
1990  003224  000000
1991                                    ;ITEM 0
1992  003226  000000  000000  000000            .WORD   0,0,0,0
1993  003234  000000
1994                                    ;ITEM 0
1995  003236  000000  000000  000000            .WORD   0,0,0,0
1996  003244  000000
1997                                    ;ITEM 0
1998  003246  000000  000000  000000            .WORD   0,0,0,0
1999  003254  000000
2000                                    ;ITEM 0
2001  003256  000000  000000  000000            .WORD   0,0,0,0
2002  003264  000000
2003                                    ;ITEM 136
2004  003266  062605  071205  073362            .WORD   EM136,DH136,DT136,DF136
2005  003274  072264
2006                                    ;ITEM 137
2007  003276  063022  071205  073362            .WORD   EM137,DH137,DT137,DF137
2008  003304  072264
2009                                    ;ITEM 140
2010  003306  063240  071252  073374            .WORD   EM140,DH140,DT140,DF140
2011  003314  072270
2012                                    ;ITEM 141
2013  003316  063601  071252  073374            .WORD   EM141,DH141,DT141,DF141
2014  003324  072270
2015                                    ;ITEM 142
2016  003326  064141  071252  073374            .WORD   EM142,DH142,DT142,DF142
2017  003334  072270
2018                                    ;ITEM 143
2019  003336  064503  071252  073374            .WORD   EM143,DH143,DT143,DF143
2020  003344  072270
2021                                    ;ITEM 144
2022  003346  065044  071252  073374            .WORD   EM144,DH144,DT144,DF144
2023  003354  072270
2024                                    ;ITEM 145
2025  003356  065376  071252  073374            .WORD   EM145,DH145,DT145,DF145
2026  003364  072270
2027                                    ;ITEM 146
2028  003366  065727  071252  073374            .WORD   EM146,DH146,DT146,DF146
2029  003374  072270
2030                                    ;ITEM 147
2031  003376  066262  071252  073374            .WORD   EM147,DH147,DT147,DF147
2032  003404  072270
2033                                    ;ITEM 150
2034  003406  066614  071315  073406            .WORD   EM150,DH150,DT150,DF150
2035  003414  072274
2036                                    ;ITEM 151
2037  003416  066700  071401  073420            .WORD   EM151,DH151,DT151,DF151
2038  003424  072300
```

```
                                                ;ITEM 152
003426  066700  071450  073420                          .WORD   EM152,DH152,DT152,DF152
003434  072300
                                                ;ITEM 153
003436  066700  071517  073420                          .WORD   EM153,DH153,DT153,DF153
003444  072300
                                                ;ITEM 154
003446  066761  071601  073430                          .WORD   EM154,DH154,DT154,DF154
003454  072303
                                                ;ITEM 155
003456  067013  071637  073430                          .WORD   EM155,DH155,DT155,DF155
003464  072303
                                                ;ITEM 156
003466  067045  071675  073430                          .WORD   EM156,DH156,DT156,DF156
003474  072303
                                                ;ITEM 0
003476  000000  000000  000000                          .WORD   0,0,0,0
003504  000000
                                                ;ITEM 160
003506  067112  071733  073420                          .WORD   EM160,DH160,DT160,DF160
003514  072303
                                                ;ITEM 161
003516  067144  071761  073420                          .WORD   EM161,DH161,DT161,DF161
003524  072303
003526  000016                                RS4REG: .WORD   16
003530  172040                                RS4CS1: .WORD   172040
003532  000000                                RS4WC:  .WORD   0
003534  000000                                RS4BA:  .WORD   0
003536  000000                                RS4DA:  .WORD   0
003540  000000                                RS4CS2: .WORD   0
003542  000000                                RS4DS:  .WORD   0
003544  000000                                RS4ER:  .WORD   0
003546  000000                                RS4AS:  .WORD   0
003550  000000                                RS4LA:  .WORD   0
003552  000000                                RS4DB:  .WORD   0
003554  000000                                RS4MR:  .WORD   0
003556  000000                                RS4DT:  .WORD   0
003560  000000                                RS4BAE: .WORD   0
003562  000000                                RS4CS3: .WORD   0

003564  000026                                RP4REG: .WORD   26
003566  176700                                RP4CS1: .WORD   176700
003570  000000                                RP4WC:  .WORD   0
003572  000000                                RP4BA:  .WORD   0
003574  000000                                RP4DA:  .WORD   0
003576  000000                                RP4CS2: .WORD   0
003600  000000                                RP4DS:  .WORD   0
003602  000000                                RP4RR1: .WORD   0
003604  000000                                RP4AS:  .WORD   0
003606  000000                                RP4LA:  .WORD   0
003610  000000                                RP4DB:  .WORD   0
003612  000000                                RP4MR:  .WORD   0
003614  000000                                RP4DT:  .WORD   0
003616  000000                                RP4SN:  .WORD   0
003620  000000                                RP4OF:  .WORD   0
003622  000000                                RP4DC:  .WORD   0
```

```
2095  003624  000000                  RP4CCC: .WORD   0
2096  003626  000000                  RP4RR2: .WORD   0
2097  003630  000000                  RP4RR3: .WORD   0
2098  003632  000000                  RP4EC1: .WORD   0
2099  003634  000000                  RP4EC2: .WORD   0
2100  003636  000000                  RP4BAE: .WORD   0
2101  003640  000000                  RP4CS3: .WORD   0
2102
2103  003642  000014                  RH4REG: .WORD   14
2104  003644  160100                  RH4CS1: .WORD   160100
2105  003646  000000                  RH4WC:  .WORD   0
2106  003650  000000                  RH4BA:  .WORD   0
2107  003652  000000                  RH4MR2: .WORD   0
2108  003654  000000                  RH4CS2: .WORD   0
2109  003656  000000                  RH4ST:  .WORD   0
2110  003660  000000                  RH4ER:  .WORD   0
2111  003662  000000                  RH4AS:  .WORD   0
2112  003664  000000                  RH4DR:  .WORD   0
2113  003666  000000                  RH4DB:  .WORD   0
2114  003670  000000                  RH4MR1: .WORD   0
2115  003672  000000                  RH4DT:  .WORD   0
2116
2117  003674  000002                  RH4REX: .WORD   2
2118  003676  160174                  RH4AE:  .WORD   160174
2119  003700  000000                  RH4CS3: .WORD   0
2120
2121  003702  000007                  RK5REG: .WORD   7
2122  003704  177400                  RK5DS:  .WORD   177400
2123  003706  000000                  RK5ER:  .WORD   0
2124  003710  000000                  RK5CS1: .WORD   0
2125  003712  000000                  RK5WC:  .WORD   0
2126  003714  000000                  RK5BA:  .WORD   0
2127  003716  000000                  RK5DA:  .WORD   0
2128  003720  000000                  RK5DB:  .WORD   0
2129
2130
2131  003722  000006                  UBEREG: .WORD   6
2132  003724  170000                  UBEDB:  .WORD   170000
2133  003726  000000                  UBECC:  .WORD   0
2134  003730  000000                  UBEBA:  .WORD   0
2135  003732  000000                  UBECR1: .WORD   0
2136  003734  000000                  UBECLR: .WORD   0
2137  003736  000000                  UBECR2: .WORD   0
2138
2139                                  ;THESE ARE THE DEVICE TRAP VECTER ADDRESSES:
2140  003740  000204                  RS4V:   .WORD   204
2141  003742  000254                  RP4V:   .WORD   254
2142  003744  000774                  RH4V:   .WORD   774
2143  003746  000220                  RK5V:   .WORD   220
2144  003750  000510                  UBEV:   .WORD   510
2145
2146
2147
2148  003752  005037  001502          START:  CLR     $TSTNM
2149                                  .SBTTL  INITIALIZE THE COMMON TAGS
2150                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
```

```
2151  003756  012706  001500                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2152  003762  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2153  003764  022706  001540                  CMP     #SWR,R6 ;;DONE?
2154  003770  001374                          BNE     .-6             ;;LOOP BACK IF NO
2155  003772  012706  001500                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2156                                  ;;INITIALIZE A FEW VECTORS
2157  003776  012737  041334  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2158  004004  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2159  004012  012737  041612  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2160  004020  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2161  004026  012737  043062  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2162  004034  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
2163  004042  012737  043160  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2164  004050  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
2165  004056  013737  041230  041222          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2166  004064  005037  001702                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2167  004070  005037  001704                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2168  004074  112737  000001  001515          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2169  004102  012737  004102  001506          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2170  004110  012737  004110  001510          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2171                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2172                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
2173  004116  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2174  004122  012737  004156  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2175  004130  012737  177570  001540          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2176  004136  012737  177570  001542          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2177  004144  022777  177777  175366          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2178  004152  001012                          BNE     65$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2179                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
2180  004154  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
2181  004156  012716  004164          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2182  004162  000002                          RTI
2183  004164  012737  000176  001540  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2184  004172  012737  000174  001542          MOV     #DISPREG,DISPLAY
2185  004200  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2186
2187                                  .SBTTL  TYPE PROGRAM NAME
2188                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2189  004204  005227  177777                  INC     #-1             ;;FIRST TIME?
2190  004210  001047                          BNE     67$             ;;BRANCH IF NO
2191  004212  022737  041300  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
2192  004220  001443                          BEQ     67$             ;;BRANCH IF YES
2193  004222  104400  004230                  TYPE    ,68$            ;;TYPE ASCIZ STRING
2194  004226  000440                          BR      67$             ;;GET OVER THE ASCIZ
2195                                  ;;68$:  .ASCIZ  <CRLF>'MAINDEC-11-DEKBD-C  PDP 11/70 CACHE MEMORY DIAGNOSTIC PART 2'<CRL
2196  004330                          67$:
2197                                  ;THIS ROUTINE SAVES THE TOP 1500 (DEC) WORDS OF THE FIRST 28K OF
2198                                  ;MEMORY. THESE LOCATIONS SHOULD CONTAIN EITHER THE MONITOR OR THE
2199                                  ;LOADER WHICH LOADED THE PROGRAM. NOTE THAT TO RESTORE THIS PART
2200                                  ;OF CORE, THAT IS TO RESTORE THE LOADER OR MONITOR, ALL THE USER
2201                                  ;MUST DO IS TYPE ↑C (CONTROL-C), WHILE THIS PROGRAM IS RUNNING.
2202                                  ;THIS WILL AUTOMATICALLY RESTORE THE TOP PART OF MEMORY TO ITS STATE
2203                                  ;BEFORE THIS PROGRAM WAS STARTED! AFTER THE MONITOR (OR LOADER) HAS BEEN
2204                                  ;RESTORED THIS PROGRAM WILL HALT.
2205  004330  005237  044234          LOOP:   INC     MONF            ;INCREMENT THE FLAG WHICH INDICATES
2206  004334  001013                          BNE     TOP             ;WHETHER OR NOT THE TOP OF MEMORY
```

```
2207
2208  004336  013737  000060  044232          MOV     @#TKVEC,MONTTY  ;IN THE FIRST 28K HAS BEEN SAVED.
2209                                                                  ;SAVE THE INITIAL CONTENTS OF THE TTY
2210  004344  012700  002734                  MOV     #↑D1500,R0      ;KEYBOARD INTERRUPT VECTOR.
2211  004350  012701  073446                  MOV     #BOTTOM+4,R1    ;IF NOT THEN SAVE IT.
2212  004354  012702  160000                  MOV     #160000,R2      ;SAVE IT AT THE BOTTOM OF THIS PROGRAM.
2213  004360  014221                  1$:     MOV     -(R2),(R1)+     ;GET THE ADDRESS OF THE END OF THE MONITOR.
2214  004362  077002                          SOB     R0,1$           ;SAVE 1500 (DEC) LOCATIONS (WORDS)
2215  004364  012737  000044  177770  TOP:    MOV     #44,@#177770
2216
2217  004372  012737  044114  000060          MOV     #RESMON,@#TKVEC ;SET THE KEYBOARD INTERRUPT VECTOR.
2218  004400  012737  000340  000062          MOV     #340,@#↑KVEC+2
2219  004406  005077  175134                  CLR     @$TKB           ;MAKE SURE THE KEYBOARD BUFFER IS CLEAR.
2220  004412  152777  000100  175124          BISB    #BIT6,@$TKS     ;TURN ON INTERRUPT ENABLE FOR THE KEYBOARD.
2221  004420  012737  043452  000004          MOV     #CPSPUR,@#4     ;SET UP FOR UNEXPECTED ERRORS.
2222  004426  012737  043500  000114          MOV     #SPUR,@#114
2223
2224                                  ;;*****************************************************************
2225                                  ;*TEST 1        CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FLOATING ONES
2226                                  ;*
2227                                  ;*THIS TEST IS A TEST OF BOTH THE AMX, CPU INPUTS, AND
2228                                  ;*THE CACHE ERROR ADDRESS REGISTER. A SET OF ADDRESSES IS
2229                                  ;*GENERATED AND A MAIN MEMORY ADDRESS AND CONTROL LINE
2230                                  ;*PARITY ERROR IS FORCED AT EACH, THEREBY LOCKING UP
2231                                  ;*THE ADDRESS ON THE OUTPUT OF  THE AMX IN THE ERROR
2232                                  ;*ADDRESS REGISTER. THE MANNER IN WHICH THIS IS DONE
2233                                  ;*IS AS FOLLOWS: FIRST THE ADDRESS IS GENERATED;
2234                                  ;*THEN, IF IT IS A VALID ADDRESS (THAT IS, IF IT IS NOT
2235                                  ;*BEYOND THE LIMITS OF MEMORY AS DISPLAYED IN THE
2236                                  ;*SYSTEM SIZE REGISTER), THESE THREE INSTRUCTIONS ARE MOVED
2237                                  ;*TO THAT AREA OF MEMORY:
2238                                  ;*      ONE:    MOV     R1,(R2)
2239                                  ;*      2$:     CLR     (R2)
2240                                  ;*      3$:     RTS     PC
2241                                  ;*2$ IS THE ADDRESS BEING TESTED. THE INSTRUCTION
2242                                  ;*AT ONE IS GIVEN CONTROL BY A 'JSR PC'. R1 IS MADE
2243                                  ;*TO CONTAIN #2 AND R2 CONTAINES THE ADDRESS OF
2244                                  ;*THE MAINTENANCE REGISTER, SO THAT AFTER THE 'MOV R1,(R2)'
2245                                  ;*IS EXECUTED A PARITY ERROR SHOULD OCCUR ON THE
2246                                  ;*MAIN MEMORY ADDRESS AND CONTROL LINES WHEN THE
2247                                  ;*NEXT INSTRUCTION IS FETCHED.
2248                                  ;*THE ADDRESSES USED ARE GENERATED FOLLOWINT THIS PATTERN
2249                                  ;*              200000
2250                                  ;*              200002
2251                                  ;*              200004
2252                                  ;*              200010
2253                                  ;*              200020
2254                                  ;*              200040
2255                                  ;*              200100
2256                                  ;*              200200
2257                                  ;*              200400
2258                                  ;*              ETC. TO:
2259                                  ;*              240000
2260                                  ;*              300000
2261                                  ;*              400000
2262                                  ;*              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.
                                        ;*
                                        ;;*****************************************************************
004434  000004                        TST1:   SCOPE
004436  012737  000020  001702                MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000001                  X=$TN-1
                                                                        ;SET THE SKAD REGISTER
004444  012737  005342  043632                MOV     #TST2,SKAD      ;IN CASE THE TEST ABORTS.

004452  113737  001502  001632                MOVB    $TSTNM,$TMPO
004460  012737  043500  000114                MOV     #SPUR,@#CACHVEC         ;INITIALLY EXPECT NO ERRORS

                                                                        ;SEE IF THIS TEST SHOULD
                                                                        ;BE EXECUTED. THE CONDITION
                                                                        ;TEST IS THE DESIGNATED
004466  104411                              MMSKIP                      ;CONSOLE SWITCH.
004470  012700  172340                      MOV     #KIPAR0,R0          ;INITIALIZE THE KERNAL
004474  012701  077406                      MOV     #77406,R1           ;SPACE MEMORY MANAGEMENT
004500  012702  172300                      MOV     #KIPDR0,R2          ;REGISTERS
004504  012703  000010                      MOV     #10,R3
004510  010122                      1$:     MOV     R1,(R2)+
004512  077302                              SOB     R3,1$
004514  005020                              CLR     (R0)+
004516  012720  000200                      MOV     #200,(R0)+
004522  012720  000400                      MOV     #400,(R0)+
004526  012720  000600                      MOV     #600,(R0)+
004532  012720  001000                      MOV     #1000,(R0)+
004536  012720  001200                      MOV     #1200,(R0)+
004542  012720  001400                      MOV     #1400,(R0)+
004546  012710  177600                      MOV     #177600,(R0)
004552  012737  000020  172516              MOV     #20,@#MMR3              ;TURN ON MEMORY MANAGEMENT
004560  012737  000001  177572              MOV     #1,@#MMR0
004566  104412                              SIZE                        ;DETERMINE FROM THE SYSTEM
                                                                        ;SIZE REGISTER WHAT THE
                                                                        ;HIGHEST ADDRESSABLE WORD
                                                                        ;OF MEMORY IS.
004570  000000                      XLOADR: .WORD   0                   ;LOW ORDER 16-BITS OF THE
004572  000000                      XHIADR: .WORD   0                   ;ADDRESS AND HIGH ORDER 6-BITS
004574  042737  000002  004570              BIC     #2,XLOADR           ;SET THE HIGHEST WORD MINUS TWO
                                                                        ;IN XLOADR.
```

```
2319
2320  004602  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
2321
2322  004610  005037  005330                  CLR     XADR3                   ;INITIALIZE STORAGE
2323  004614  005037  005332                  CLR     XADR3+2                 ;LOCATIONS USED TO GENERATE
2324  004620  005037  005320                  CLR     XADR1                   ;THE SERIES OF TEST ADDRESSES.
2325  004624  012737  000001  005322          MOV     #1,XADR1+2
2326
2327  004632                          X1:
2328
2329                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2330  004632  023737  005322  005332          CMP     XADR1+2,XADR3+2 ;COMPARE THE HIGH ORDER
2331  004640  001006                          BNE     64$             ;PARTS OF XADR1 AND ARG2.
2332  004642  023737  005320  005330          CMP     XADR1,XADR3     ;COMPARE THE LOW ORDER
2333  004650  001002                          BNE     64$             ;PARTS.
2334
2335
2336
2337  004652  000137  005274                  JMP     X11             ;THEY WERE EQUAL!
2338
2339  004656  103402                  64$:    BLO     65$
2340  004660  000137  004670                  JMP     X2              ;THE FIRST ADDRESS IS LARGER
2341                                                                  ;THAN THE SECOND!
2342  004664  000137  005274          65$:    JMP     X11             ;THE FIRST IS LESS THAN THE
2343                                                                  ;SECOND.
2344
2345
2346  004670                          X2:
2347                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
2348  004670  013737  005320  005324          MOV     XADR1,XADR2
2349  004676  013737  005322  005326          MOV     XADR1+2,XADR2+2
2350  004704  063737  005330  005324          ADD     XADR3,XADR2
2351  004712  005537  005326                  ADC     XADR2+2
2352  004716  063737  005332  005326          ADD     XADR3+2,XADR2+2
2353
2354
2355
2356
2357  004724                          X3:
2358
2359                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2360  004724  023737  005326  004572          CMP     XADR2+2,XLOADR+2        ;COMPARE THE HIGH ORDER
2361  004732  001006                          BNE     64$             ;PARTS OF XADR2 AND ARG2.
2362  004734  023737  005324  004570          CMP     XADR2,XLOADR    ;COMPARE THE LOW ORDER
2363  004742  001002                          BNE     64$             ;PARTS.
2364
2365
2366
2367  004744  000137  005340                  JMP     XDONE           ;THEY WERE EQUAL!
2368
2369  004750  103402                  64$:    BLO     65$
2370  004752  000137  005340                  JMP     XDONE           ;THE FIRST ADDRESS IS LARGER
2371                                                                  ;THAN THE SECOND!
2372  004756  000137  004762          65$:    JMP     X4              ;THE FIRST IS LESS THAN THE
2373                                                                  ;SECOND.
2374
```

```
2375  004762  012737  004762  001510  X4:     MOV     #X4,$LPERR
2376
2377                                    ;CONVERT THE 22-BIT ADDRESS IN XADR2 TO VIRTUAL ADDRESS
2378                                    ;WHICH WILL RELOCATE THROUGH KIPAR6; SET UP KIPAR6;
2379                                    ;TURN ON MEMORY MANAGEMENT; PUT THE INSTRUCTIONS:
2380                                    ;       1$:     MOV     R1,(R2)
2381                                    ;       2$:     CLR     (R2)
2382                                    ;       3$:     RTS     PC
2383                                    ;AT THE LOCATION BEING TESTED, WITH 2$=TEST ADDRESS;
2384                                    ;PUT A PATTERN,000002, IN R1 FOR THE MAINTENANCE
2385                                    ;REGISTER TO FORCE BAD PARITY ON THE MAIN MEMORY
2386                                    ;ADDRESS AND CONTROL LINES.  PUT THE ADDRESS OF
2387                                    ;THE CACHE MAINTENANCE REGISTER IN R2.  PUT THE
2388                                    ;ADDRESS, X6, IN LOCATION CACHVEC TO TAKE CARE OF THE
2389                                    ;WHICH IS BEING FORCED. JSR TO THE ABOVE ROUTINE,
2390                                    ;SO THAT IF THE PARITY ERROR DOES'NT OCCUR
2391                                    ;THE 'RTS PC', AT 3$ ABOVE, WILL HANDLE IT.
2392
2393  004770  013703  005324                    MOV     XADR2,R3
2394  004774  013702  005326                    MOV     XADR2+2,R2
2395  005000  162703  000002                    SUB     #2,R3
2396  005004  005602                            SBC     R2
2397
2398  005006  010300                            MOV     R3,R0
2399  005010  042700  177701                    BIC     #177701,R0
2400  005014  062700  140000                    ADD     #140000,R0
2401  005020  073227  177772                    ASHC    #-6,R2
2402  005024  010337  172354                    MOV     R3,@#KIPAR6
2403
2404  005030  012737  000020  172516            MOV     #20,@#MMR3      ;TURN ON MEMORY
2405  005036  012737  000001  177572            MOV     #1,@#MMR0       ;MANAGEMENT.
2406                                            ;SET UP THE TEST INSTRUCTIONS.
2407  005044  012710  010112                    MOV     #010112,(R0)    ;010112 = 'MOV R1,(R2)'
2408  005050  012760  005012  000002            MOV     #005012,2(R0)   ;005012 = 'CLR (R2)'
2409  005056  012760  000207  000004            MOV     #000207,4(R0)   ;000207 = 'RTS PC'
2410
2411  005064  012701  000002                    MOV     #2,R1           ;SET UP THE REGISTERS
2412  005070  012702  177750                    MOV     #MAINT,R2
2413
2414  005074  012737  005114  000114            MOV     #X6,@#CACHVEC   ;SET UP THE PARITY ERROR
2415  005102  000240                            NOP                     ;TRAP VECTOR AND GO.
2416  005104  004710                            JSR     PC,(R0)
2417
2418  005106                          X5:                             ;NO TRAP OR ABORT OCCURRED!
2419                                                                          ;MAINTENANCE FUNCTION
2420                                                                          ;FOR BAD PARITY ON
2421  005106  104022                  1$:     ERROR   22                      ;THE MAIN MEMORY ADDRESS
2422  005110  000137  005226                    JMP     X9                      ;AND CONTROL LINES FAILED
2423
2424  005114                          X6:
2425
2426                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2427  005114  023737  005326  177742            CMP     XADR2+2,LOADRS+2        ;COMPARE THE HIGH ORDER
2428  005122  001006                            BNE     64$             ;PARTS OF XADR2 AND ARG2.
2429  005124  023737  005324  177740            CMP     XADR2,LOADRS    ;COMPARE THE LOW ORDER
2430  005132  001002                            BNE     64$             ;PARTS.
```

```
2431
2432
2433
2434  005134  000137  005152              JMP     X7              ;THEY WERE EQUAL!
2435
2436  005140  103402          64$:        BLO     65$
2437  005142  000137  005170              JMP     X8              ;THE FIRST ADDRESS IS LARGER
2438                                                              ;THAN THE SECOND!
2439  005146  000137  005170      65$:    JMP     X8              ;THE FIRST IS LESS THAN THE
2440                                                              ;SECOND.
2441
2442                                                                      ;PARITY ERROR OCCURS.
2443  005152  005726              X7:     TST     (SP)+                   ;RESTORE THE STACK.
2444  005154  022626                      CMP     (SP)+,(SP)+             ;AND CONTINUE SINCE
2445  005156  012737  177777  177744      MOV     #-1,@#MEMERR            ;THE CACHE ERROR ADDRESS
2446  005164  000137  005226              JMP     X9                      ;REGISTER WAS SET CORRECTLY.
2447
2448  005170  013737  177744  001634  X8: MOV     @#MEMERR,$TMP1          ;REPORT VALID TEST
2449                                                                      ;FAILURE.
2450  005176  013737  177740  001640      MOV     @#LOADRS,$TMP3
2451  005204  013737  177742  001642      MOV     @#HIADRS,$TMP4
2452  005212  005726                      TST     (SP)+
2453  005214  022626                      CMP     (SP)+,(SP)+
2454  005216  104023                      ERROR   23
2455  005220  012737  177777  177744      MOV     #-1,@#MEMERR
2456
2457  005226  005037  177572          X9: CLR     @#MMR0                  ;TURN OFF MEMORY MANAGEMENT.
2458  005232  005037  172516              CLR     @#MMR3
2459  005236  005737  005330              TST     XADR3
2460  005242  001007                      BNE     X10                     ;GET READY TO GENERATE
2461  005244  005737  005332              TST     XADR3+2                 ;THE NEXT TEST ADDRESS.
2462  005250  001004                      BNE     X10
2463  005252  012737  000002  005330      MOV     #2,XADR3
2464  005260  000415                      BR      X12
2465  005262  006337  005330      X10:    ASL     XADR3
2466  005266  006137  005332              ROL     XADR3+2
2467  005272  000410                      BR      X12
2468
2469  005274  006337  005320      X11:    ASL     XADR1
2470  005300  006137  005322              ROL     XADR1+2
2471  005304  005037  005330              CLR     XADR3
2472  005310  005037  005332              CLR     XADR3+2
2473  005314  000137  004632      X12:    JMP     X1
2474
2475  005320  000000              XADR1:  .WORD   0
2476  005322  000000                      .WORD   0
2477  005324  000000              XADR2:  .WORD   0
2478  005326  000000                      .WORD   0
2479  005330  000000              XADR3:  .WORD   0
2480  005332  000000                      .WORD   0
2481  005334  000000              XADR4:  .WORD   0
2482  005336  000000                      .WORD   0
2483  005340  104407              XDONE:  RSET                            ;DONE!
2484
2485                              ;;*****************************************************************
2486                              ;*TEST 2         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.
                                ;*
                                ;;*****************************************************************
005342  000004               TST2:   SCOPE
005344  012737  000020  001702        MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000002               XX=$TN-1
                                                             ;SET THE SKAD REGISTER
005352  012737  006242  043632        MOV     #TST3,SKAD      ;IN CASE THE TEST ABORTS.

005360  113737  001502  001632        MOVB    $TSTNM,$TMPO
005366  012737  043500  000114        MOV     #SPUR,@#CACHVEC         ;INITIALLY EXPECT NO ERRORS.

005374  104411                       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.
005376  012700  172340               MOV     #KIPAR0,R0              ;INITIALIZE THE KERNAL MODE
005402  012701  077406               MOV     #77406,R1               ;MEMORY MANAGEMENT REGISTERS.
005406  012702  172300               MOV     #KIPDR0,R2
005412  012703  000010               MOV     #10,R3
005416  010122               1$:     MOV     R1,(R2)+
005420  077302                       SOB     R3,1$
005422  005020                       CLR     (R0)+
005424  012720  000200               MOV     #200,(R0)+
005430  012720  000400               MOV     #400,(R0)+
005434  012720  000600               MOV     #600,(R0)+
005440  012720  001000               MOV     #1000,(R0)+
005444  012720  001200               MOV     #1200,(R0)+
005450  012720  001400               MOV     #1400,(R0)+
005454  012710  177600               MOV     #177600,(R0)
005460  012737  000020  172516       MOV     #20,@#MMR3              ;TRUN ON MEMORY MANAGEMENT
005466  012737  000001  177572       MOV     #1,@#MMR0
005474  104412                       SIZE                    ;GET THE LARGEST MEMORY
005476  000000               XXLOA:  .WORD   0                       ;WORD ADDRESS INTO XXLOA
005500  000000               XXHIA:  .WORD   0                       ;AND XXHIA.
005502  042737  000002  005476       BIC     #2,XXLOA                ;GET THE ADDRESS OF THE HIGHEST WORD
                                                                     ;WORD MINUS TWO.
```

```
2543
2544  005510  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FROM NOW ON FORCE MISSES
2545                                                                          ;TO BOTH GROUPS.
2546
2547  005516  012737  177776  006220  XX1:    MOV     #177776,XXADR1          ;INITIALIZE
2548  005524  005037  006222                  CLR     XXADR1+2
2549  005530  012704  000016                  MOV     #16,R4
2550  005534  000410                          BR      XX3
2551
2552  005536  005204                  XX2:    INC     R4                      ;TURN ON THE NEXT BIT
2553  005540  052737  000001  006220          BIS     #1,XXADR1               ;IN THE FIELD OF ONES.
2554  005546  006337  006220                  ASL     XXADR1
2555  005552  006137  006222                  ROL     XXADR1+2
2556
2557  005556  012737  000002  006230  XX3:    MOV     #2,XXMASK               ;INITIALIZE THE MASK
2558  005564  005037  006232                  CLR     XXMASK+2                ;USED TO CREATE THE ZERO
2559                                                                          ;IN THE FIELD OF ONES.
2560  005570  010405                          MOV     R4,R5
2561  005572  012737  005600  001510          MOV     #XX4,$LPERR
2562
2563  005600  013737  006220  006224  XX4:    MOV     XXADR1,XXADR2           ;DETERMINE THIS TEST ADDRESS.
2564  005606  013737  006222  006226          MOV     XXADR1+2,XXADR2+2
2565  005614  043737  006230  006224          BIC     XXMASK,XXADR2
2566  005622  043737  006232  006226          BIC     XXMASK+2,XXADR2+2
2567
2568
2569                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2570  005630  023737  006226  006236          CMP     XXADR2+2,XXCNST+2       ;COMPARE THE HIGH ORDER
2571  005636  001006                          BNE     64$                     ;PARTS OF XXADR2 AND ARG2.
2572  005640  023737  006224  006234          CMP     XXADR2,XXCNST           ;COMPARE THE LOW ORDER
2573  005646  001002                          BNE     64$                     ;PARTS.
2574
2575
2576
2577  005650  000137  005666                  JMP     XX5                     ;THEY WERE EQUAL!
2578
2579  005654  103402                  64$:    BLO     65$
2580  005656  000137  005666                  JMP     XX5                     ;THE FIRST ADDRESS IS LARGER
2581                                                                          ;THAN THE SECOND!
2582  005662  000137  006156          65$:    JMP     XX10                    ;THE FIRST IS LESS THAN THE
2583                                                                          ;SECOND.
2584
2585
2586  005666                          XX5:
2587
2588                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2589  005666  023737  006226  005500          CMP     XXADR2+2,XXLOA+2        ;COMPARE THE HIGH ORDER
2590  005674  001006                          BNE     64$                     ;PARTS OF XXADR2 AND ARG2.
2591  005676  023737  006224  005476          CMP     XXADR2,XXLOA            ;COMPARE THE LOW ORDER
2592  005704  001002                          BNE     64$                     ;PARTS.
2593
2594
2595
2596  005706  000137  005724                  JMP     XX6                     ;THEY WERE EQUAL!
2597
2598  005712  103402                  64$:    BLO     65$
```

```
2599  005714  000137  006156              JMP     XX10            ;THE FIRST ADDRESS IS LARGER
2600                                                              ;THAN THE SECOND!
2601  005720  000137  005724      65$:    JMP     XX6             ;THE FIRST IS LESS THAN THE
2602                                                              ;SECOND.
2603
2604
2605  005724                      XX6:
2606
2607                              ;CONVERT THE 22-BIT ADDRESS IN XXADR2 TO VIRTUAL ADDRESS
2608                              ;WHICH WILL RELOCATE THROUGH KIPAR6; SET UP KIPAR6;
2609                              ;TURN ON MEMORY MANAGEMENT; PUT THE INSTRUCTIONS:
2610                              ;       1$:     MOV     R1,(R2)
2611                              ;       2$:     CLR     (R2)
2612                              ;       3$:     RTS     PC
2613                              ;AT THE LOCATION BEING TESTED, WITH 2$=TEST ADDRESS;
2614                              ;PUT A PATTERN,000002, IN R1 FOR THE MAINTENANCE
2615                              ;REGISTER TO FORCE BAD PARITY ON THE MAIN MEMORY
2616                              ;ADDRESS AND CONTROL LINES.  PUT THE ADDRESS OF
2617                              ;THE CACHE MAINTENANCE REGISTER IN R2.  PUT THE
2618                              ;ADDRESS, XX7, IN LOCATION CACHVEC TO TAKE CARE OF THE
2619                              ;WHICH IS BEING FORCED. JSR TO THE ABOVE ROUTINE,
2620                              ;SO THAT IF THE PARITY ERROR DOES'NT OCCUR
2621                              ;THE 'RTS PC', AT 3$ ABOVE, WILL HANDLE IT.
2622
2623  005724  013703  006224              MOV     XXADR2,R3
2624  005730  013702  006226              MOV     XXADR2+2,R2
2625  005734  162703  000002              SUB     #2,R3
2626  005740  005602                      SBC     R2
2627
2628  005742  010300                      MOV     R3,R0
2629  005744  042700  177701              BIC     #177701,R0
2630  005750  062700  140000              ADD     #140000,R0
2631  005754  073227  177772              ASHC    #-6,R2
2632  005760  010337  172354              MOV     R3,@#KIPAR6
2633
2634  005764  012737  000020  172516      MOV     #20,@#MMR3      ;TURN ON MEMORY
2635  005772  012737  000001  177572      MOV     #1,@#MMR0       ;MANAGEMENT.
2636                                                              ;SET UP THE TEST INSTRUCTIONS.
2637  006000  012710  010112              MOV     #010112,(R0)    ;010112 = 'MOV R1,(R2)'
2638  006004  012760  005012  000002      MOV     #005012,2(R0)   ;005012 = 'CLR (R2)'
2639  006012  012760  000207  000004      MOV     #000207,4(R0)   ;000207 = 'RTS PC'
2640
2641  006020  012701  000002              MOV     #2,R1           ;SET UP THE REGISTERS
2642  006024  012702  177750              MOV     #MAINT,R2
2643
2644  006030  012737  006046  000114      MOV     #XX7,@#CACHVEC  ;SET UP THE PARITY ERROR
2645  006036  000240                      NOP                     ;TRAP VECTOR AND GO.
2646  006040  004710                      JSR     PC,(R0)
2647
2648                                                              ;NO TRAP OCCURRED!
2649  006042  104024              1$:     ERROR   24
2650  006044  000444                      BR      XX10
2651                              ;COME HERE ON THE PARITY ERROR
2652  006046                      XX7:
2653
2654                              ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
```

```
2655  006046  023737  006226  177742          CMP     XXADR2+2,LOADRS+2       ;COMPARE THE HIGH ORDER
2656  006054  001006                          BNE     64$             ;PARTS OF XXADR2 AND ARG2.
2657  006056  023737  006224  177740          CMP     XXADR2,LOADRS   ;COMPARE THE LOW ORDER
2658  006064  001002                          BNE     64$             ;PARTS.
2659
2660
2661
2662  006066  000137  006104                  JMP     XX8             ;THEY WERE EQUAL!
2663
2664  006072  103402                  64$:    BLO     65$
2665  006074  000137  006120                  JMP     XX9             ;THE FIRST ADDRESS IS LARGER
2666                                                                  ;THAN THE SECOND!
2667  006100  000137  006120          65$:    JMP     XX9             ;THE FIRST IS LESS THAN THE
2668                                                                  ;SECOND.
2669
2670
2671  006104  005726                  XX8:    TST     (SP)+                   ;RESTORE THE STACK.
2672  006106  022626                          CMP     (SP)+,(SP)+
2673  006110  012737  177777  177744          MOV     #-1,@#MEMERR            ;RESET THE CACHE ERROR REGISTERS.
2674  006116  000417                          BR      XX10
2675  006120  013737  177744  001634  XX9:    MOV     @#MEMERR,$TMP1          ;REPORT A VALID TEST
2676                                                                  ;FAILURE.
2677  006126  013737  177740  001640          MOV     @#LOADRS,$TMP3
2678  006134  013737  177742  001642          MOV     @#HIADRS,$TMP4
2679  006142  005726                          TST     (SP)+
2680  006144  022626                          CMP     (SP)+,(SP)+
2681  006146  104025                          ERROR   25
2682  006150  012737  177777  177744          MOV     #-1,@#MEMERR
2683
2684  006156  006337  006230          XX10:   ASL     XXMASK                  ;ROTATE THE MASK.
2685  006162  006137  006232                  ROL     XXMASK+2
2686  006166  005305                          DEC     R5
2687  006170  001402                          BEQ     1$
2688  006172  000137  005600                  JMP     XX4
2689  006176  005037  177572          1$:     CLR     @#MMR0                  ;TURN OF MEMORY MANAGEMENT.
2690  006202  005037  172516                  CLR     @#MMR3
2691  006206  020427  000025                  CMP     R4,#25
2692  006212  002012                          BGE     XX11
2693  006214  000137  005536                  JMP     XX2
2694
2695  006220  000000          XXADR1: .WORD   0                       ;USED TO GENERATE TEST PATTERNS.
2696  006222  000000                  .WORD   0
2697  006224  000000          XXADR2: .WORD   0                       ;USED TO STORE THE CURRENT
2698  006226  000000                  .WORD   0                       ;TEST PATTERN DURING A TEST.
2699  006230  000000          XXMASK: .WORD   0                       ;MASK USED TO PUT A ZERO
2700  006232  000000                  .WORD   0                       ;IN THE FIELD OF ONES
2701                                                                  ;TO CREATE A TEST ADDRESS.
2702  006234  101442          XXCNST: .WORD   BOTFRG                  ;THE SMALLEST ADDRESS
2703  006236  000000                  .WORD   0                       ;IN MEMORY OVER THIS TEST.
2704
2705  006240  104407          XX11:   RSET
2706
2707                          ;;*****************************************************************
2708                          ;*TEST 3        CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ONES
2709                          ;*
2710                          ;*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.
                                        ;*
                                        ;;*****************************************************************
006242  000004                TST3:   SCOPE
006244  012737  000020  001702        MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000003          RR=$TN-1
                                                                ;SET THE SKAD REGISTER
006252  012737  007154  043632        MOV     #TST4,SKAD      ;IN CASE THE TEST ABORTS.

006260  113737  001502  001632        MOVB    $TSTNM,$TMP0
006266  012737  043500  000114        MOV     #SPUR,@#CACHVEC         ;INITIALLY EXPECT NO ERRORS.
006274  012737  043452  000004        MOV     #CPSPUR,@#ERRVEC

006302  104411                        MMSKIP

006304  012700  172340                MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
006310  012701  077406                MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
006314  012702  172300                MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
006320  012703  000010                MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
006324  010122          64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
006326  077302                  SOB     R3,64$          ;ADDRESSES.
006330  005020                  CLR     (R0)+
006332  012720  000200          MOV     #200,(R0)+
006336  012720  000400          MOV     #400,(R0)+
006342  012720  000600          MOV     #600,(R0)+
006346  012720  001000          MOV     #1000,(R0)+
006352  012720  001200          MOV     #1200,(R0)+
006356  012720  001400          MOV     #1400,(R0)+
006362  012710  177600          MOV     #177600,(R0)

006366  012737  000060  172516        MOV     #60,@#MMR3              ;TURN ON MEMORY MANAGEMENT.
006374  012737  000001  177572        MOV     #1,@#MMR0
                                                                ;DETERMINE THE MEMORY
006402  104412                        SIZE                    ;SYSTEM SIZE.
006404  000000          RRLOAD: .WORD   0               ;LOW ORDER 16-BITS AND
006406  000000          RRHIAD: .WORD   0               ;HIGH ORDER 6-BITS OF THE
                                                                ;HIGHEST MEMORY WORD ADDRESS.
006410  042737  000002  006404        BIC     #2,RRLOAD               ;GET THE HIGHEST WORD IN MEMORY
                                                                ;MINUS TWO.
006416  012737  000014  177746        MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS

006424  005037  007146                CLR     RRADR3                  ;INITIALIZE STORAGE LOCATIONS
006430  005037  007150                CLR     RRADR3+2                ;USED TO GENERATE THE
006434  005037  007136                CLR     RRADR1          ;SERIES OF TEST ADDRESSES.
006440  012737  000001  007140        MOV     #1,RRADR1+2

006446                          RR1:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
006446  023737  007140  007150        CMP     RRADR1+2,RRADR3+2       ;COMPARE THE HIGH ORDER
```

```
2767  006454  001006                        BNE     64$             ;PARTS OF RRADR1 AND ARG2.
2768  006456  023737  007136  007146        CMP     RRADR1,RRADR3   ;COMPARE THE LOW ORDER
2769  006464  001002                        BNE     64$             ;PARTS.
2770
2771
2772
2773  006466  000137  007112                JMP     RR11            ;THEY WERE EQUAL!
2774
2775  006472  103402                64$:    BLO     65$
2776  006474  000137  006504                JMP     RR2             ;THE FIRST ADDRESS IS LARGER
2777                                                                ;THAN THE SECOND!
2778  006500  000137  007112          65$:  JMP     RR11            ;THE FIRST IS LESS THAN THE
2779                                                                ;SECOND.
2780
2781
2782  006504                          RR2:
2783                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
2784  006504  013737  007136  007142        MOV     RRADR1,RRADR2
2785  006512  013737  007140  007144        MOV     RRADR1+2,RRADR2+2
2786  006520  063737  007146  007142        ADD     RRADR3,RRADR2
2787  006526  005537  007144              ADC     RRADR2+2
2788  006532  063737  007150  007144        ADD     RRADR3+2,RRADR2+2
2789
2790
2791
2792
2793  006540                          RR3:
2794
2795                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2796  006540  023737  007144  006406        CMP     RRADR2+2,RRLOAD+2       ;COMPARE THE HIGH ORDER
2797  006546  001006                        BNE     64$             ;PARTS OF RRADR2 AND ARG2.
2798  006550  023737  007142  006404        CMP     RRADR2,RRLOAD   ;COMPARE THE LOW ORDER
2799  006556  001002                        BNE     64$             ;PARTS.
2800
2801
2802
2803  006560  000137  007152                JMP     RRDONE          ;THEY WERE EQUAL!
2804
2805  006564  103402                64$:    BLO     65$
2606  006566  000137  007152                JMP     RRDONE          ;THE FIRST ADDRESS IS LARGER
2807                                                                ;THAN THE SECOND!
2808  006572  000137  006576          65$:  JMP     RR4             ;THE FIRST IS LESS THAN THE
2809                                                                ;SECOND.
2810
2811  006576  012737  006576  001510  RR4:   MOV     #RR4,$LPERR
2812                                  ;CONVERT THE PHYSICAL 22-BIT, ADDRESS IN RRADR2 TO A VIRTUAL ADDRESS
2813                                  ;WHICH WILL RELOCATE THROUGH KIPAR6 TO THE UNIBUS, THEN THROUGH
2814                                  ;THE MAPPING BOX TO THE UNIBUS INPUTS OF THE CACHE AMX.
2815  006604  013737  007142  170200        MOV     RRADR2,@#MAPLO0 ;SET UP THE MAP REGISTER 0.
2816  006612  013737  007144  170202        MOV     RRADR2+2,@#MAPHI0
2817  006620  162737  000002  170200        SUB     #2,@#MAPLO0
2818  006626  005637  170202              SBC     @#MAPHI0
2819
2820  006632  012700  140000                MOV     #140000,R0      ;A VIRTUAL ADDRESS WHICH WILL
2821                                                                ;RELOCATE THROUGH KIPAR6.
2822  006636  012737  170000  172354        MOV     #170000,@#KIPAR6;RELOCATE TO UNIBUS BASE
```

```
2823                                                            ;ADDRESS OF 000000.
2824  006644  012737  000060  172516        MOV     #60,@#MMR3    ;TURN ON THE MAPPING BOX AND
2825                                                            ;22-BIT MODE.
2826  006652  012737  000001  177572        MOV     #1,@#MMR0     ;TURN ON MEMORY MANAGEMENT.
2827                                                            ;SET UP THE TEST CODE:
2828  006660  012710  010112               MOV     #010112,(R0)  ;010112='MOV R1,(R2)'
2829  006664  012760  005012  000002        MOV     #005012,2(R0) ;005012='CLR (R2)'
2830  006672  012760  000207  000004        MOV     #000207,4(R0) ;000207='RTS PC'
2831
2832  006700  012701  000002               MOV     #2,R1         ;SET UP THE REGISTERS USED
2833  006704  012702  177750               MOV     #MAINT,R2     ;IN THE TEST INSTRUCTIONS.
2834
2835  006710  012737  006730  000114        MOV     #RR6,@#CACHVEC ;SET UP THE PARITY TRAP
2836  006716  000240                       NOP                   ;VECTOR.
2837  006720  004710                       JSR     PC,(R0) ;AND GO.
2838
2839
2840  006722                       RR5:                          ;NO TRAP OR ABORT OCCURRED!
2841                                                                     ;MAINTENANCE FUNCTION FOR
2842  006722  104030               1$:     ERROR   30                    ;FORCING BAD PARITY ON
2843  006724  000137  007044               JMP     RR9                   ;THE MAIN MEMORY ADDRESS
2844                                                                     ;AND CONTROL LINES FAILED.
2845                                ;COME HERE WHEN THE FORCED ERROR OCCURS.
2846  006730                       RR6:
2847
2848                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2849  006730  023737  007144  177742        CMP     RRADR2+2,LOADRS+2     ;COMPARE THE HIGH ORDER
2850  006736  001006                       BNE     64$           ;PARTS OF RRADR2 AND ARG2.
2851  006740  023737  007142  177740        CMP     RRADR2,LOADRS ;COMPARE THE LOW ORDER
2852  006746  001002                       BNE     64$           ;PARTS.
2853
2854
2855
2856  006750  000137  006766               JMP     RR7           ;THEY WERE EQUAL!
2857
2858  006754  103402               64$:    BLO     65$
2859  006756  000137  007006               JMP     RR8           ;THE FIRST ADDRESS IS LARGER
2860                                                            ;THAN THE SECOND!
2861  006762  000137  007006               JMP     RR8           ;THE FIRST IS LESS THAN THE
2862                                                            ;SECOND.
2863
2864
2865  006766  022626               RR7:    CMP     (SP)+,(SP)+
2866  006770  005726                       TST     (SP)+                 ;RESTORE THE STACK.
2867  006772  022626                       CMP     (SP)+,(SP)+
2868  006774  012737  177777  177744        MOV     #-1,@#MEMERR          ;CLEAR THE CACHE ERROR REGISTER.
2869  007002  000137  007044               JMP     RR9
2870
2871  007006  013737  177744  001634  RR8:  MOV     @#MEMERR,$TMP1        ;REPORT A VALID TEST FAILURE.
2872  007014  013737  177740  001640        MOV     @#LOADRS,$TMP3
2873  007022  013737  177742  001642        MOV     @#HIADRS,$TMP4
2874  007030  005726                       TST     (SP)+
2875  007032  022626                       CMP     (SP)+,(SP)+
2876  007034  104031                       ERROR   31
2877  007036  012737  000001  177744        MOV     #1,@#MEMERR           ;CLEAR THE ERROR REGISTER.
2878  007044  005037  177572       RR9:    CLR     @#MMR0                ;TURN OFF MEMORY MANAGEMENT.
```

```
 2879  007050  005037  172516                    CLR     @#MMR3
 2880  007054  005737  007146                    TST     RRADR3                  ;GET READY TO GENERATE THE
 2881  007060  001007                            BNE     RR10                    ;NEXT ADDRESS TO BE TESTED.
 2882  007062  005737  007146                    TST     RRADR3
 2883  007066  001004                            BNE     RR10
 2884  007070  012737  000002  007146            MOV     #2,RRADR3
 2885  007076  000415                            BR      RR12
 2886
 2887  007100  006337  007146          RR10:     ASL     RRADR3
 2888  007104  006137  007150                    ROL     RRADR3+2
 2889  007110  000410                            BR      RR12
 2890
 2891  007112  006337  007136          RR11:     ASL     RRADR1
 2892  007116  006137  007140                    ROL     RRADR1+2
 2893  007122  005037  007146                    CLR     RRADR3
 2894  007126  005037  007150                    CLR     RRADR3+2
 2895
 2896  007132  000137  006446          RR12:     JMP     RR1
 2897
 2898  007136  000000                  RRADR1:   .WORD   0                       ;3 DOUBLE WORD LOCATIONS
 2899  007140  000000                            .WORD   0                       ;USED TO STORE 22-BIT
 2900  007142  000000                  RRADR2:   .WORD   0                       ;ADDRESSES.
 2901  007144  000000                            .WORD   0
 2902  007146  000000                  RRADR3:   .WORD   0
 2903  007150  000000                            .WORD   0
 2904
 2905  007152  104407                  RRDONE:   RSET                            ;DONE!
 2906
 2907                                            ;;*****************************************************************
 2908                                            ;*TEST 4        CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ZEROES
 2909                                            ;*
 2910                                            ;*THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX.
 2911                                            ;*THIS TEST IS IDENTICAL TO TST2 IN EVERY THING
 2912                                            ;*IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE
 2913                                            ;*REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM
 2914                                            ;*THE CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL
 2915                                            ;*GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS
 2916                                            ;*WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE
 2917                                            ;*AS UNIBUS REFERENCES.
 2918                                            ;*
 2919                                            ;;*****************************************************************
 2920  007154  000004                  TST4:     SCOPE
 2921  007156  012737  000020  001702            MOV     #20,$TIMES              ;;DO 20 ITERATIONS
 2922           000004                  SS=$TN-1
 2923                                                                            ;SET THE SKAD REGISTER
 2924  007164  012737  010042  043632            MOV     #TST5,SKAD              ;IN CASE THE TEST ABORTS.
 2925
 2926  007172  113737  001502  001632            MOVB    $TSTNM,$TMP0
 2927  007200  012737  043500  000114            MOV     #SPUR,@#CACHVEC                 ;INITIALLY EXPECT NO ERRORS
 2928  007206  104411                            MMSKIP
 2929
 2930  007210  012700  172340                    MOV     #KIPAR0,R0              ;INITIALLY PUT MEMORY
 2931  007214  012701  077406                    MOV     #77406,R1               ;MANAGEMENT IN A 'PASSIVE'
 2932  007220  012702  172300                    MOV     #KIPDR0,R2              ;STATE, THAT IS MAP ALL
 2933  007224  012703  000010                    MOV     #10,R3                  ;VIRTUAL ADDRESSES ON TO
 2934  007230  010122                  64$:      MOV     R1,(R2)+                ;THEMSELVES AS PHYSICAL
```

```
2935  007232  077302                          SUB     R3,64$          ;ADDRESSES.
2936  007234  005020                          CLR     (R0)+
2937  007236  012720  000200                  MOV     #200,(R0)+
2938  007242  012720  000400                  MOV     #400,(R0)+
2939  007246  012720  000600                  MOV     #600,(R0)+
2940  007252  012720  001000                  MOV     #1000,(R0)+
2941  007256  012720  001200                  MOV     #1200,(R0)+
2942  007262  012720  001400                  MOV     #1400,(R0)+
2943  007266  012710  177600                  MOV     #177600,(R0)
2944
2945  007272  104412                          SIZE                            ;GET THE MEMORY SIZE.
2946  007274  000000                  SSLOAD: .WORD   0                       ;22-BIT ADDRESS OF THE
2947  007276  000000                  SSHIAD: .WORD   0                       ;HIGHEST WORD IN MEMORY.
2948  007300  042737  000002  007274          BIC     #2,SSLOAD               ;GET THE HIGHEST WORD MINUS TWO.
2949
2950  007306  012737  000014  177746          MOV     #MOM1,@#CONTRL
2951
2952  007314  012737  177776  010020  SS1:    MOV     #177776,SSADR1          ;INITIALIZE
2953  007322  005037  010022                  CLR     SSADR1+2
2954  007326  012704  000016                  MOV     #16,R4
2955  007332  000410                          BR      SS3
2956
2957  007334  005204                  SS2:    INC     R4                      ;TURN ON THE NEXT BIT
2958  007336  052737  000001  010020          BIS     #1,SSADR1               ;IN THE FIELD OF ONES
2959  007344  006337  010020                  ASL     SSADR1
2960  007350  006137  010022                  ROL     SSADR1+2
2961
2962  007354  012737  000002  010030  SS3:    MOV     #2,SSMASK               ;INITIALIZE THE MASK USER
2963  007362  005037  010032                  CLR     SSMASK+2                ;TO CREATE THE ZERO IN
2964                                                                          ;IN FIELD OF ONES
2965  007366  010405                          MOV     R4,R5
2966  007370  012737  007376  001510          MOV     #SS4,$LPERR
2967
2968  007376  013737  010020  010024  SS4:    MOV     SSADR1,SSADR2           ;DETERMINE THE TEST ADDRESS.
2969  007404  013737  010022  010026          MOV     SSADR1+2,SSADR2+2
2970  007412  043737  010030  010024          BIC     SSMASK,SSADR2
2971  007420  043737  010032  010026          BIC     SSMASK+2,SSADR2+2
2972
2973                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
2974  007426  023737  010026  010036          CMP     SSADR2+2,SSCNST+2       ;COMPARE THE HIGH ORDER
2975  007434  001006                          BNE     64$             ;PARTS OF SSADR2 AND ARG2.
2976  007436  023737  010024  010034          CMP     SSADR2,SSCNST   ;COMPARE THE LOW ORDER
2977  007444  001002                          BNE     64$             ;PARTS.
2978
2979
2980
2981  007446  000137  007464                  JMP     SS5             ;THEY WERE EQUAL!
2982
2983  007452  103402                  64$:    BLO     65$
2984  007454  000137  007464                  JMP     SS5             ;THE FIRST ADDRESS IS LARGER
2985                                                                  ;THAN THE SECOND!
2986  007460  000137  007756          65$:    JMP     SS10            ;THE FIRST IS LESS THAN THE
2987                                                                  ;SECOND.
2988
2989  007464                          SS5:
2990
```

```
                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007464  023737  010026  007276            CMP     SSADR2+2,SSLOAD+2       ;COMPARE THE HIGH ORDER
007472  001006                            BNE     64$                     ;PARTS OF SSADR2 AND ARG2.
007474  023737  010024  007274            CMP     SSADR2,SSLOAD           ;COMPARE THE LOW ORDER
007502  001002                            BNE     64$                     ;PARTS.


007504  000137  007522                    JMP     SS6                     ;THEY WERE EQUAL!

007510  103402                  64$:      BLO     65$
007512  000137  007756                    JMP     SS10                    ;THE FIRST ADDRESS IS LARGER
                                                                          ;THAN THE SECOND!
007516  000137  007522          65$:      JMP     SS6                     ;THE FIRST IS LESS THAN THE
                                                                          ;SECOND.


007522                          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.
007522  013737  010024  170200            MOV     SSADR2,@#MAPLOO ;SET UP THE MAP REGISTER 0.
007530  013737  010026  170202            MOV     SSADR2+2,@#MAPHOO
007536  162737  000002  170200            SUB     #2,@#MAPLOO
007544  005637  170202                    SBC     @#MAPHOO

007550  012700  140000                    MOV     #140000,R0              ;A VIRTUAL ADDRESS WHICH WILL
                                                                          ;RELOCATE THROUGH KIPAR6.
007554  012737  170000  172354            MOV     #170000,@#KIPAR6;RELOCATE TO UNIBUS BASE
                                                                          ;ADDRESS OF 000000.
007562  012737  000060  172516            MOV     #60,@#MMR3              ;TURN ON THE MAPPING BOX AND
                                                                          ;22-BIT MODE.
007570  012737  000001  177572            MOV     #1,@#MMR0               ;TURN ON MEMORY MANAGEMENT.
                                                                          ;SET UP THE TEST CODE:
007576  012710  010112                    MOV     #010112,(R0)            ;010112='MOV R1,(R2)'
007602  012760  005012  000002            MOV     #005012,2(R0)           ;005012='CLR (R2)'
007610  012760  000207  000004            MOV     #000207,4(R0)           ;000207='RTS PC'

007616  012701  000002                    MOV     #2,R1                   ;SET UP THE REGISTERS USED
007622  012702  177750                    MOV     #MAINT,R2               ;IN THE TEST INSTRUCTIONS.

007626  012737  007702  000114            MOV     #SS8,@#CACHVEC  ;SET UP THE PARITY TRAP
007634  000240                            NOP                             ;VECTOR.
007636  004710                            JSR     PC,(R0) ;AND GO.

                                                                          ;NO TRAP OCCURRED!
007640  104032                  1$:       ERROR   32
007642  000445                            BR      SS10
                                ;TRAP TO HERE WHEN THE ERROR OCCURS.
007644                          SS7:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007644  023737  010026  177742            CMP     SSADR2+2,LOADRS+2       ;COMPARE THE HIGH ORDER
007652  001006                            BNE     64$                     ;PARTS OF SSADR2 AND ARG2.
007654  023737  010024  177740            CMP     SSADR2,LOADRS           ;COMPARE THE LOW ORDER
007662  001002                            BNE     64$                     ;PARTS.
```

```
3047
3048
3049
3050  007664  000137  007702                  JMP     SS8             ;THEY WERE EQUAL!
3051
3052  007670  103402                  64$:    BLO     65$
3053  007672  000137  007720                  JMP     SS9             ;THE FIRST ADDRESS IS LARGER
3054                                                                    ;THAN THE SECOND!
3055  007676  000137  007720          65$:    JMP     SS9             ;THE FIRST IS LESS THAN THE
3056                                                                    ;SECOND.
3057
3058
3059  007702  022626                  SS8:    CMP     (SP)+,(SP)+
3060  007704  005726                          TST     (SP)+                   ;RESTORE THE STACK
3061  007706  022626                          CMP     (SP)+,(SP)+
3062  007710  012737  177777  177744          MOV     #-1,@#MEMERR            ;CLEAR THE CACHE ERROR
3063  007716  000417                          BR      SS10                    ;REGISTER.
3064
3065  007720  013737  177744  001634  SS9:    MOV     @#MEMERR,$TMP1          ;REPORT A VALID TEST FAILURE.
3066  007726  013737  177740  001640          MOV     @#LOADRS,$TMP3
3067  007734  013737  177742  001642          MOV     @#HIADRS,$TMP4
3068  007742  005726                          TST     (SP)+
3069  007744  022626                          CMP     (SP)+,(SP)+
3070  007746  104033                          ERROR   33
3071  007750  012737  177777  177744          MOV     #-1,@#MEMERR
3072
3073  007756  006337  010030          SS10:   ASL     SSMASK                  ;ROTATE MASK TO FLOAT 0
3074  007762  006137  010032                  ROL     SSMASK+2                ;TO THE LEFT.
3075  007766  005305                          DEC     R5
3076  007770  001402                          BEQ     1$
3077  007772  000137  007376                  JMP     SS4
3078  007776  005037  177572          1$:     CLR     @#MMR0                  ;TURN OF MEMORY MANAGEMENT
3079  010002  005037  172516                  CLR     @#MMR3          ;AND THE MAPPING BOX.
3080  010006  020427  000025                  CMP     R4,#25          ;IS THE TEST DONE?
3081  010012  002012                          BGE     SS11            ;YES
3082  010014  000137  007334                  JMP     SS2             ;NO
3083
3084  010020  000000          SSADR1: .WORD   0               ;USED TO GENERATE THE
3085  010022  000000                  .WORD   0               ;TEST ADDRESSESS.
3086  010024  000000          SSADR2: .WORD   0
3087  010026  000000                  .WORD   0
3088  010030  000000          SSMASK: .WORD   0
3089  010032  000000                  .WORD   0
3090
3091  010034  101442          SSCNST: .WORD   BOTPRG           ;CONTAINS THE ADDRESS OF
3092  010036  000000                  .WORD   0               ;THE LAST WORD OF THIS PROGRAM.
3093
3094  010040  104407          SS11:   RSET                    ;DONE!
3095
3096                          ;;*************************************************************
3097                          ;*TEST 5        CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS DUAL ADDRESS TEST
3098                          ;*
3099                          ;*THIS TEST PERFORMS A DUAL ADDRESS TEST ON MEMORY LOCATED
3100                          ;*AT ADDRESSES LESS THAN 160000 (OCT.) OR WITHIN THE FIRST
3101                          ;*28K.  THE PURPOSE IS TO VARIFY THE THE AMX IS WORKING
3102                          ;*PROPERLY FOR THE LOW ORDER ADDRESS LINES INVOLVED.
```

```
3103                                    ;*
3104                                    ;;*****************************************************************
3105  010042  000004                    TST5:   SCOPE
3106  010044  012737  000004  001702            MOV     #4,$TIMES       ;;DO 4 ITERATIONS
3107           000005                   PP=$TN-1
3108                                                                    ;SET THE SKAD REGISTER
3109  010052  012737  010300  043632            MOV     #TST6,SKAD      ;IN CASE THE TEST ABORTS.
3110
3111  010060  113737  001502  001632            MOVB    $TSTNM,$TMPO
3112  010066  012737  043500  000114            MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
3113
3114  010074  012737  000014  177746  PP1:      MOV     #M1MO,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS
3115  010102  104412                            SIZE
3116  010104  000000                   PPLOAD: .WORD   0               ;LOW ORDER 16-BITS AND
3117  010106  000000                   PPHIAD: .WORD   0               ;HIGH ORDER 6-BITS OF THE
3118                                                                    ;HIGHEST WORD ADDRESS IN
3119                                                                    ;MEMORY.
3120  010110  012737  157776  010274            MOV     #157776,PPLIM   ;ESTABLISH THE UPPER LIMIT
3121  010116  005737  010106                    TST     PPHIAD          ;FOR THE TEST.
3122  010122  001007                            BNE     PP2
3123  010124  023737  010274  010104            CMP     PPLIM,PPLOAD
3124  010132  003403                            BLE     PP2
3125  010134  013737  010104  010274            MOV     PPLOAD,PPLIM
3126
3127  010142  012700  101442          PP2:      MOV     #BOTPRG,R0      ;THE LOW LIMIT FOR THIS TEST.
3128  010146  010020                   1$:       MOV     R0,(R0)+        ;WRITE THE ADDRESS IN THE
3129  010150  020037  010274                    CMP     R0,PPLIM        ;ADDRESS.
3130  010154  101774                            BLOS    1$
3131
3132  010156  012700  101442                    MOV     #BOTPRG,R0
3133  010162  011001                   PP3:      MOV     (R0),R1         ;GO BACK AND READ BACK THE
3134  010164  020001                             CMP     R0,R1           ;ADDRESS, CHECK IT AND
3135  010166  001411                             BEQ     PP4             ;WRITE BACK THE COMPLIMENT.
3136  010170  010037  001644                    MOV     R0,$TMP5
3137                                                                    ;REPORT ERROR.
3138  010174  010137  001636                    MOV     R1,$TMP2
3139  010200  010037  001640                    MOV     R0,$TMP3
3140  010204  005037  001642                    CLR     $TMP4
3141  010210  104034                   1$:       ERROR   34
3142
3143  010212  005120                   PP4:      COM     (R0)+           ;WRITE BACK COMPLIMENT.
3144  010214  020037  010274                    CMP     R0,PPLIM
3145  010220  101760                            BLOS    PP3
3146
3147  010222  012700  101442                    MOV     #BOTPRG,R0      ;GO BACK AND CHECK
3148  010226  011001                   PP5:      MOV     (R0),R1         ;THE COMPLIMENTED PATTERNS.
3149  010230  010002                             MOV     R0,R2
3150  010232  005102                             COM     R2
3151  010234  020102                             CMP     R1,R2
3152  010236  001411                             BEQ     PP6
3153  010240  010237  001644                    MOV     R2,$TMP5
3154  010244  010137  001636                    MOV     R1,$TMP2
3155  010250  010037  001640                    MOV     R0,$TMP3
3156  010254  005037  001642                    CLR     $TMP4
3157  010260  104034                   1$:       ERROR   34
3158
```

```
3159  010262  005120                  PP6:    CUM     (R0)+
3160  010264  020037  010274                  CMP     R0,PPLIM
3161  010270  001356                  PP5     BNE     PP5
3162  010272  000401                          BR      PP7
3163
3164  010274  000000                  PPLIM:  .WORD   0
3165
3166  010276  104407                  PP7:    RSET                    ;DONE!
3167
3168
3169                                  ;;*****************************************************************
3170                                  ;*TEST 6         CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS DUAL ADDRESS TEST
3171                                  ;*
3172                                  ;*THIS TEST PERFORMS A DUAL ADDRESS TEST IDENTICAL TO
3173                                  ;*TST5, EXCEPT THAT IT IS DONE THROUGH THE MAPPING
3174                                  ;*BOX HERE THEREBY TESTING THE UNIBUS INPUTS TO THE AMX.
3175                                  ;*
3176                                  ;;*****************************************************************
3177  010300  000004                  TST6:   SCOPE
3178  010302  012737  000002  001702          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
3179           000006                 TT=$TN-1
3180                                                                  ;SET THE SKAD REGISTER
3181  010310  012737  010740  043632          MOV     #TST7,SKAD      ;IN CASE THE TEST ABORTS.
3182
3183  010316  113737  001502  001632          MOVB    $TSTNM,$TMP0
3184  010324  012737  043500  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
3185  010332  104411                          MMSKIP
3186  010334  012737  000014  177746  TT1:    MOV     #M1MO,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
3187  010342  104412                          SIZE
3188  010344  000000                  TTLOAD: .WORD   0               ;DETERMINE THE HIGHEST
3189  010346  000000                  TTHIAD: .WORD   0               ;WORD IN MEMORY.
3190
3191  010350  012737  157776  010734          MOV     #157776,TTLIM   ;DETERMINE THE UPPER LIMIT
3192  010356  005737  010346                  TST     TTHIAD          ;FOR THE TEST.
3193  010362  001007                          BNE     TT2
3194  010364  023737  010734  010344          CMP     TTLIM,TTLOAD
3195  010372  003403                          BLE     TT2
3196  010374  013737  010344  010734          MOV     TTLOAD,TTLIM
3197  010402                  TT2:
3198
3199  010402  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
3200  010406  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
3201  010412  012702  172300                  MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
3202  010416  012703  000010                  MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
3203  010422  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
3204  010424  077302                          SOB     R3,64$          ;ADDRESSES.
3205  010426  005020                          CLR     (R0)+
3206  010430  012720  000200                  MOV     #200,(R0)+
3207  010434  012720  000400                  MOV     #400,(R0)+
3208  010440  012720  000600                  MOV     #600,(R0)+
3209  010444  012720  001000                  MOV     #1000,(R0)+
3210  010450  012720  001200                  MOV     #1200,(R0)+
3211  010454  012720  001400                  MOV     #1400,(R0)+
3212  010460  012710  177600                  MOV     #177600,(R0)
3213
3214  010464  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON MEMORY MANAGEMENT.
```

```
3215  010472  012737  000001  177572          MOV     #1,@#MMR0
3216  010500  012700  101442                  MOV     #BOTPRG,R0      ;INITIALIZE A POINTER.
3217
3218  010504                          1$:
3219
3220  010504  010037  170200                  MOV     R0,@#MAPLO0     ;RELOCATE THE ADDRESS IN
3221  010510  005037  170202                  CLR     @#MAPHO0        ;R0 TO THE UNIBUS,
3222  010514  012737  170000  172354          MOV     #170000,@#KIPAR6;THROUGH THE MAPPING BOX
3223  010522  012701  140000                  MOV     #140000,R1      ;TO THE CACHE.
3224
3225
3226  010526  010011                          MOV     R0,(R1)         ;WRITE THE ADDRESS IN THE
3227  010530  062700  000002                  ADD     #2,R0           ;ADDRESS
3228  010534  020037  010734                  CMP     R0,TTLIM
3229  010540  101761                          BLOS    1$
3230
3231  010542  012700  101442                  MOV     #BOTPRG,R0
3232
3233  010546                          TT3:
3234
3235  010546  010037  170200                  MOV     R0,@#MAPLO0     ;RELOCATE THE ADDRESS IN
3236  010552  005037  170202                  CLR     @#MAPHO0        ;R0 TO THE UNIBUS,
3237  010556  012737  170000  172354          MOV     #170000,@#KIPAR6;THROUGH THE MAPPING BOX
3238  010564  012701  140000                  MOV     #140000,R1      ;TO THE CACHE.
3239
3240
3241  010570  011102                          MOV     (R1),R2         ;READ BACK THE ADDRESS
3242  010572  020002                          CMP     R0,R2           ;AS DATA IN THE LOCATION
3243  010574  001411                          BEQ     TT4             ;IT ADDRESSES.
3244  010576  010037  001644                  MOV     R0,$TMP5        ;REPORT ERROR IF NOT
3245                                                                  ;EQUAL.
3246  010602  010237  001636                  MOV     R2,$TMP2
3247  010606  010037  001640                  MOV     R0,$TMP3
3248  010612  005037  001642                  CLR     $TMP4
3249  010616  104035                  1$:     ERROR   35
3250  010620  005111                  TT4:    COM     (R1)            ;WRITE BACK THE
3251  010622  062700  000002                  ADD     #2,R0           ;COMPLIMENTED DATA.
3252  010626  020037  010734                  CMP     R0,TTLIM
3253  010632  101745                          BLOS    TT3
3254
3255  010634  012700  101442                  MOV     #BOTPRG,R0
3256
3257  010640                          TT5:
3258
3259  010640  010037  170200                  MOV     R0,@#MAPLO0     ;RELOCATE THE ADDRESS IN
3260  010644  005037  170202                  CLR     @#MAPHO0        ;R0 TO THE UNIBUS,
3261  010650  012737  170000  172354          MOV     #170000,@#KIPAR6;THROUGH THE MAPPING BOX
3262  010656  012701  140000                  MOV     #140000,R1      ;TO THE CACHE.
3263
3264
3265  010662  011102                          MOV     (R1),R2         ;GO BACK AND CHECK
3266  010664  010003                          MOV     R0,R3           ;THE COMPLIMENTED PATTERNS.
3267  010666  005103                          COM     R3
3268  010670  020203                          CMP     R2,R3
3269  010672  001411                          BEQ     TT6
3270  010674  010337  001644                  MOV     R3,$TMP5        ;REPORT ERROR
```

```
3271  010700  010237  001636                MOV     R2,$TMP2
3272  010704  010037  001640                MOV     R0,$TMP3
3273  010710  005037  001642                CLR     $TMP4
3274  010714  104035                1$:     ERROR   35
3275
3276  010716  005111                TT6:    COM     (R1)            ;COMPLIMENT BACK THE DATA.
3277  010720  062700  000002                ADD     #2,R0
3278  010724  020037  010734                CMP     R0,TTLIM
3279  010730  001343                        BNE     TT5
3280  010732  000401                        BR      TT7
3281
3282  010734  000000                TTLIM:  .WORD   0
3283
3284  010736  104407                TT7:    RSET                    ;DONE!
3285
3286
3287                                        ;;*****************************************************************
3288                                        ;*TEST 7        CACHE ADDRESS MEMORY COMPARATOR TEST
3289                                        ;*
3290                                        ;*THIS IS A TEST OF THE CACHE ADDRESS MEMORY ADDRESS COMPARATORS.
3291                                        ;*THIS IS A CIRCUIT MADE UP OF SIX 74585 CHIPS, THREE FOR EACH
3292                                        ;*GROUP.  EACH CHIP COMPARES FOUR BITS OF THE ADDRESS ON THE
3293                                        ;*ADDRESS MULTIPLEXER, AMX, OUTPUT LINES WITH THE RESPECTIVE
3294                                        ;*FOUR BITS FROM THE CACHE ADDRESS MEMORY.  TWELVE BITS OF
3295                                        ;*THE ADDRESS ARE BROKEN DOWN THUS:  BITS 10 THROUGH 13
3296                                        ;*FOR THE FIRST COMPARATOR; BITS 14 THROUGH 17 FOR
3297                                        ;*THE NEXT; AND BITS 18 THROUGH 21 FOR THE LAST.
3298                                        ;*THE METHOD CHOSEN FOR THIS TEST IS TO TAKE EACH
3299                                        ;*POSSIBLE 4-BIT INPUT CONDITION FOR A COMPARATOR FROM THE
3300                                        ;*ADDRESS MEMORY AND PUT EVERY POSSIBLE 4-BIT COMBINATION
3301                                        ;*ON THE AMX SIDE OF THE COMPARATOR.  FOR 4-BITS
3302                                        ;*THERE ARE 16 (DEC) CONDITIONS.  THUS FOR EVERY 4-BIT
3303                                        ;*ADDRESS MEMORY INPUT TO THE COMPARATOR THERE ARE
3304                                        ;*16 AMX INPUT COMBINATIONS ONE OF WHICH WILL CAUSE
3305                                        ;*A MATCH AND MAKE THE REFERENCE A HIT.  THE OTHER
3306                                        ;*15 SHOULD OF COURSE BE MISSES.
3307                                        ;*
3308                                        ;*****************************************************************
3309  010740  000004                TST7:   SCOPE
3310  010742  012737  000040  001702        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3311                                                                ;SET THE SKAD REGISTER
3312  010750  012737  012112  043632        MOV     #TST10,SKAD     ;IN CASE THE TEST ABORTS.
3313
3314  010756  113737  001502  001632        MOVB    $TSTNM,$TMP0
3315  010764  012737  043500  000114        MOV     #SPUR,@#CACHVEC
3316
3317  010772  104411                        MMSKIP                  ;SEE IF THE SWITCH REGISTER
3318                                                                ;REFLECTS THE USERS DESIRE
3319                                                                ;TO ELIMINATE EXECUTION OF ANY  ESTS
3320                                                                ;USING MEMORY MANAGEMENT.  IF
3321                                                                ;SO GO TO THE NEXT TEST.
3322
3323  010774  012700  172340                MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
3324  011000  012701  077406                MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
3325  011004  012702  172300                MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
3326  011010  012703  000010                MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
```

```
3327  011014  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
3328  011016  077302                          SOB     R3,64$          ;ADDRESSES.
3329  011020  005020                          CLR     (R0)+
3330  011022  012720  000200                MOV     #200,(R0)+
3331  011026  012720  000400                MOV     #400,(R0)+
3332  011032  012720  000600                MOV     #600,(R0)+
3333  011036  012720  001000                MOV     #1000,(R0)+
3334  011042  012720  001200                MOV     #1200,(R0)+
3335  011046  012720  001400                MOV     #1400,(R0)+
3336  011052  012710  177600                MOV     #177600,(R0)
3337
3338
3339  011056  104412                  SIZE
3340  011060  000000                  ZADLO:  .WORD   0               ;THE HIGHEST ADDRESSABLE
3341  011062  000000                  ZADHI:  .WORD   0               ;MEMORY WORD AVAILABLE.
3342
3343  011064  005037  011656                  CLR     ZFLG1           ;ZFLG1 INDICATES WHICH GROUP
3344                                                                  ;IS BEING TESTED.
3345                                                                  ;ZFLG1 = 0, TESTING GROUP 0.
3346                                                                  ;ZFLG1 = 1, TESTING GROUP 1.
3347                                                                  ;TEST GROUP 0 FIRST.
3348
3349  011070  012737  000030  011664          MOV     #SOM1,ZGS       ;ZGS AND ZGM CONTAIN
3350  011076  012737  000044  011662          MOV     #S1M0,ZGM       ;PATTERNS TO BE USED IN
3351                                                                  ;THE CACHE CONTROL REGISTER.
3352  011104  005037  011660                  CLR     ZFLG2           ;ZFLG2 INDICATES WHICH
3353                                                                  ;4-BIT ADDRESS FIELD, OR
3354                                                                  ;WHICH COMPARATOR, IS
3355                                                                  ;BEING TESTED.
3356                                                                  ;ZFLG2 = 0, BITS 10 THROUGH 13
3357                                                                  ;ZFLG2 = 1, BITS 14 THROUGH 17
3358                                                                  ;ZFLG2 = 2, BITS 18 THROUGH 21
3359                                                                  ;ZFLG2 = 3, DONE!
3360
3361  011110  005737  011660          Z1:     TST     ZFLG2           ;SEE WHICH COMPARATOR
3362  011114  001010                          BNE     Z2              ;IS BEING TESTED ON THIS
3363                                                                  ;PASS AND PUT THE SIXTEEN
3364                                                                  ;POSSIBLE ADDRESSES NEEDED
3365                                                                  ;FOR THE TEST IN ZTABLE.
3366  011116  012737  002000  011704          MOV     #2000,ZTABLE+4  ;BITS 10-13
3367  011124  005037  011706                  CLR     ZTABLE+6
3368  011130  004737  012002                  JSR     PC,ZCMTBL       ;CALL ZCMTBL TO FINISH THE TABLE.
3369  011134  000432                          BR      Z5
3370
3371  011136  022737  000001  011660  Z2:     CMP     #1,ZFLG2
3372  011144  001010                          BNE     Z3
3373
3374  011146  012737  040000  011704          MOV     #40000,ZTABLE+4 ;BITS 14-17
3375  011154  005037  011706                  CLR     ZTABLE+6
3376  011160  004737  012002                  JSR     PC,ZCMTBL       ;GET ZCMTBL TO FINISH SETTING
3377  011164  000416                          BR      Z5              ;UP THE TABLE.
3378
3379  011166  022737  000002  011660  Z3:     CMP     #2,ZFLG2
3380  011174  001010                          BNE     Z4
3381
3382  011176  012737  000004  011706          MOV     #4,ZTABLE+6     ;BITS 18-21
```

```
3383  011204  005037  011704          CLR     ZTABLE+4
3384  011210  004737  012002          JSR     PC,ZCMTBL
3385  011214  000402                  ER      Z5
3386
3387  011216  000137  011610  Z4:     JMP     Z14             ;DONE WITH THIS GROUP.
3388
3389  011222  012701  011670  Z5:     MOV     #ZTHR,R1
3390  011226  013737  011662  177746  MOV     ZGM,@#CONTRL
3391  011234  005711                  TST     (R1)            ;MAKE ZTHR A HIT IN BOTH GROUPS.
3392  011236  013737  011664  177746  MOV     ZGS,@#CONTRL
3393  011244  005711                  TST     (R1)
3394
3395                                                          ;FROM NOW ON SELECT THE GROUP BEING TESTED
3396                                                          ;WHILE MISSING THE OTHER GROUP.
3397  011246  012737  000020  172516  MOV     #20,@#MMR3      ;TURN ON MEMORY MANAGEMENT.
3398  011254  012737  000001  177572  MOV     #1,@#MMR0       ;22-BIT MODE!
3399
3400  011262  012701  011700          MOV     #ZTABLE,R1      ;INITIALIZE R1 AS A POINTER
3401                                                          ;TO THE ADDRESS WHICH WILL
3402                                                          ;BE MADE A HIT.
3403
3404  011266                  Z7:
3405
3406                          ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
3407
3408
3409  011266  023761  011062  000002          CMP     ZADLO+2,2(R1)   ;COMPARE THE HIGH ORDER
3410  011274  001005                          BNE     64$             ;PARTS OF ZADLO AND (R1).
3411  011276  023711  011060                  CMP     ZADLO,(R1)      ;THEN IF NECESSARY
3412  011302  001002                          BNE     64$             ;COMPARE THE LOW ORDER PARTS.
3413
3414  011304  000137  011322                  JMP     1$              ;THEY WERE EQUAL!
3415
3416  011310  103402          64$:    BLO     65$
3417  011312  000137  011322                  JMP     1$              ;THE FIRST ADDRESS IS LARGER
3418                                                                  ;THAN THE SECOND!
3419  011316  000137  011610          65$:    JMP     Z14             ;THE FIRST IS LESS THAN THE
3420                                                                  ;SECOND.
3421
3422
3423  011322  012702  011700          1$:     MOV     #ZTABLE,R2      ;INITIALIZE A POINTER TO
3424                                                                  ;THE ADDRESSES WHICH WILL
3425                                                                  ;BE FED THROUGH THE COMPARATOR
3426                                                                  ;AGAINST THE ADDRESS POINTED
3427                                                                  ;TO BY THE OTHER POINTER, R1
3428
3429  011326  020102          Z8:     CMP     R1,R2           ;DON'T TEST THE ADDRESS
3430  011330  001511                  BEQ     Z12             ;AGAINST ITSELF HERE.
3431
3432  011332                  Z9:
3433
3434                          ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
3435
3436
3437  011332  023762  011062  000002          CMP     ZADLO+2,2(R2)   ;COMPARE THE HIGH ORDER
3438  011340  001005                          BNE     64$             ;PARTS OF ZADLO AND (R2).
```

```
3439  011342  023712  011060          CMP     ZADLO,(R2)      ;THEN IF NECESSARY
3440  011346  001002                  BNE     64$             ;COMPARE THE LOW ORDER PARTS.
3441
3442  011350  000137  011366          JMP     Z10             ;THEY WERE EQUAL!
3443
3444  011354  103402          64$:    BLO     65$
3445  011356  000137  011366          JMP     Z10             ;THE FIRST ADDRESS IS LARGER
3446                                                          ;THAN THE SECOND!
3447  011362  000137  011566  65$:    JMP     Z13             ;THE FIRST IS LESS THAN THE
3448                                                          ;SECOND.
3449
3450
3451  011366          Z10:
3452
3453  011366  011103                  MOV     (R1),R3         ;GET THE PHYSICAL ADDRESS POINTED
3454  011370  042703  177700          BIC     #177700,R3      ;TO BY R1 AND ESTABLISH
3455  011374  011105                  MOV     (R1),R5 ;A VIRTUAL ADDRESS WHICH
3456  011376  016104  000002          MOV     2(R1),R4        ;WILL RELOCATE THROUGH
3457  011402  073427  177772          ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
3458  011406  010537  172354          MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
3459  011412  062703  140000          ADD     #140000,R3      ;IN R3.
3460
3461
3462  011416  005713                  TST     (R3)
3463  011420  005713                  TST     (R3)            ;SEE IF YOU CAN GET A HIT.
3464  011422  032737  000010  177752  BIT     #10,@#HITMIS
3465  011430  001011                  BNE     Z11
3466  011432  013737  011656  001634  MOV     ZFLG:,$TMP1     ;NO! REPORT THE FAILURE
3467  011440  011137  001636          MOV     (R1),$TMP2
3468  011444  016137  000002  001640  MOV     2(R1),$TMP3
3469  011452  104026          1$:     ERROR   26
3470
3471  011454          Z11:
3472
3473  011454  011203                  MOV     (R2),R3         ;GET THE PHYSICAL ADDRESS POINTED
3474  011456  042703  177700          BIC     #177700,R3      ;TO BY R2 AND ESTABLISH
3475  011462  011205                  MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
3476  011464  016204  000002          MOV     2(R2),R4        ;WILL RELOCATE THROUGH
3477  011470  073427  177772          ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
3478  011474  010537  172354          MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
3479  011500  062703  140000          ADD     #140000,R3      ;IN R3.
3480
3481
3482  011504  000240                  NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
3483  011506  005713                  TST     (R3)            ;MAKE SURE THERE IS NO
3484  011510  032737  000010  177752  BIT     #10,@#HITMIS            ;MATCH.  A MISS?
3485  011516  001416                  BEQ     Z12
3486  011520  013737  011656  001634  MOV     ZFLG1,$TMP1     ;GOT A HIT! SO REPORT
3487  011526  011137  001636          MOV     (R1),$TMP2      ;FAILURE
3488  011532  016137  000002  001640  MOV     2(R1),$TMP3
3489  011540  011237  001642          MOV     (R2),$TMP4
3490  011544  016237  000002  001644  MOV     2(R2),$TMP5
3491  011552  104027          1$:     ERROR   27
3492
3493  011554  062702  000004  Z12:    ADD     #4,R2           ;MOVE POINTER TO NEXT AMX
3494                                                          ;SIDE COMPARATOR INPUT ADDRESS.
```

```
3495  011560  020227  012000              CMP     R2,#ZTABOT      ;DONE?
3496  011564  001260                      BNE     Z8              ;BRANCH IF NOT DONE.
3497
3498  011566  062701  000004      Z13:    ADD     #4,R1           ;GO TO THE NEXT ADDRESS
3499  011572  020127  012000              CMP     R1,#ZTABOT      ;IN THE TABLE; OR IS THE
3500  011576  001233                      BNE     Z7              ;TEST USING THIS ADDRESS TABLE DONE?
3501                                                              ;IF NOT GO TO Z7.
3502  011600  005237  011660              INC     ZFLG2           ;IF DONE WITH THESE ADDRESSES
3503  011604  000137  011110              JMP     Z1              ;GO BACK TO COMPUTE THE
3504                                                              ;NEXT ADDRESS TABLE, THAT IS
3505                                                              ;CHECK THE NEXT 4-BIT
3506                                                              ;COMPARATOR
3507  011610  005037  177572      Z14:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
3508  011614  005037  172516              CLR     @#MMR3
3509  011620  005737  011656              TST     ZFLG1           ;SEE IF BOTH GROUPS HAVE
3510  011624  001131                      BNE     Z15             ;BEEN TESTED. BRANCH IF YES
3511  011626  005237  011656              INC     ZFLG1           ;OTHERWISE CHANGE THE
3512  011632  012737  000044  011664      MOV     #S1M0,ZGS       ;PATTERNS USED IN THE CACHE
3513  011640  012737  000030  011662      MOV     #S0M1,ZGM       ;CONTROL REGISTER AND GO
3514  011646  005037  011660              CLR     ZFLG2           ;BACK TO TEST GROUP 1.
3515  011652  000137  011110              JMP     Z1
3516
3517  011656  000000              ZFLG1:  .WORD   0               ;FLAG WHICH DESIGNATES WHICH
3518                                                              ;GROUP IS BEING TESTED, 0 OR 1.
3519  011660  000000              ZFLG2:  .WORD   0               ;FLAG WHICH DESIGNATES WHICH
3520                                                              ;COMPARATOR IS BEING TESTED:
3521                                                              ;0 - BITS 10 THROUGH 13
3522                                                              ;1 - BITS 14 THROUGH 17
3523                                                              ;2 - BITS 18 THROUGH 21.
3524
3525  011662  000000              ZGM:    .WORD   0               ;PATTERNS USED IN THE HIT
3526  011664  000000              ZGS:    .WORD   0               ;AND MISS REGISTER.
3527  011666  000000                      .WORD   0
3528  011670  000000              ZTHR:   .WORD   0
3529  011672  000000                      .WORD   0
3530
3531  011674  000000              ZTMP1:  .WORD   0               ;TEMPORARY STORAGE LOCATIONS
3532  011676  000000              ZTMP2:  .WORD   0               ;USED BY THE ROUTINE, ZCMTBL,
3533                                                              ;TO GENERATE THE TEST ADDRESS
3534                                                              ;TABLE, ZTABLE.
3535
3536  011700  000040              ZTABLE: .BLKW   40              ;THE TEST ADDRESS TABLE.
3537  012000  000000              ZTABOT: .WORD   0               ;PRECISION, 22-BIT, ADDRESSES.
3538
3539                              ;THIS ROUTINE IS CALLED TO GENERATE THE TEST ADDRESS
3540                              ;TABLE, BY A 'JSR PC,ZCMTBL'.  IT CLEARS THE FIRST
3541                              ;ENTRY; IT ASSUMES THE THE BASE ADDRESS HAS BEEN
3542                              ;PLACED IN THE SECOND ENTRY BEFORE CONTROL IS PASSED
3543                              ;TO IT; THEN, STARTING WITH THE THIRD ENTRY, IT COMPUTES
3544                              ;EACH ENTRY BY ADDING THE BASE ADDRESS TO THE PRECEEDING
3545                              ;ENTRY.
3546  012002  012701  011700      ZCMTBL: MOV     #ZTABLE,R1      ;ESTABLISH A POINTER TO
3547                                                              ;THE TABLE.
3548  012006  005021                      CLR     (R1)+           ;CLR THE FIRST ENTRY.
3549  012010  005021                      CLR     (R1)+
3550  012012  012700  000016              MOV     #16,R0
```

```
3551  012016  012137  011674          1$:     MOV     (R1)+,ZTMP1     ;SAVE THE CURRENT ENTRY
3552  012022  012137  011676                  MOV     (R1)+,ZTMP2
3553                                                                    ;ADD THE OFFSET TO THE
3554                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
3555
3556
3557
3558  012026  013711  011674                  MOV     ZTMP1,(R1)
3559  012032  013761  011676  000002          MOV     ZTMP1+2,2(R1)
3560  012040  063711  011704                  ADD     ZTABLE+4,(R1)
3561  012044  005561  000002                  ADC     2(R1)
3562  012050  063761  011706  000002          ADD     ZTABLE+4+2,2(R1)
3563  012056  077021                          SOB     R0,1$           ;LOOP UNTIL ZTABLE IS FILLED.
3564
3565
3566  012060  012702  000020                  MOV     #20,R2
3567  012064  012701  011700                  MOV     #ZTABLE,R1
3568  012070  012700  011670                  MOV     #ZTHR,R0
3569  012074  042700  176000                  BIC     #176000,R0
3570  012100  060021                  2$:     ADD     R0,(R1)+
3571  012102  005721                          TST     (R1)+
3572  012104  077203                          SOB     R2,2$
3573
3574  012106  000207                          RTS     PC              ;THE RETURN
3575
3576  012110  104407                  Z15:    RSET                    ;DONE!
3577
3578
3579                                  ;;**************************************************************
3580                                  ;*TEST 10       CACHE ADDRESS MEMORY COUNT PATTERN TEST
3581                                  ;*
3582                                  ;*THIS IS A TEST OF THE ADDRESS MEMORY IN THE CACHE.
3583                                  ;*EVERY BIT IN THE MEMORY IS TURNED ON AND OFF WITHIN
3584                                  ;*THE LIMITATIONS OF MEMORY SIZE.  THE MANNER IN WHICH
3585                                  ;*THIS IS DONE IS TO ATTEMPT TO MAKE EVERY ADDRESS
3586                                  ;*IN AVAILABLE MEMORY A HIT IN EACH GROUP.
3587                                  ;*
3588                                  ;;**************************************************************
3589  012112  000004                  TST10:  SCOPE
3590  012114  012737  000002  001702          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
3591          000010                  BB=$TN-1
3592  012122                          BB0:
3593                                                                  ;SET THE SKAD REGISTER
3594  012122  012737  013142  043632          MOV     #TST11,SKAD     ;IN CASE THE TEST ABORTS.
3595
3596  012130  113737  001502  001632          MOVB    $TSTNM,$TMP0
3597
3598  012136  104411                          MMSKIP
3599
3600  012140  104412                          SIZE
3601  012142  000000                  BBLOAD: .WORD   0
3602  012144  000000                  BBHIAD: .WORD   0
3603
3604  012146  005037  012644                  CLR     BBFLG1          ;TEST GROUP 0 FIRST.
3605  012152  012737  000034  012654          MOV     #SOMOM1,BBGS
3606  012160  012737  000054  012656          MOV     #S1MOM1,BBGM
```

```
3607
3608  012166  012737  043500  000114  BB1:    MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS, FOR NOW.
3609  012174  012700  012122                  MOV     #BB0,R0         ;MAKE THIS CODE HITS IN
3610  012200  012701  001000                  MOV     #1000,R1        ;THE GROUP NOT BEING TESTED.
3611  012204  013737  012654  177746  BB2:    MOV     BBGS,@#CONTRL
3612  012212  005760  002000                  TST     2000(R0)
3613  012216  013737  012656  177746          MOV     BBGM,@#CONTRL
3614  012224  005720                          TST     (R0)+
3615  012226  077112                          SOB     R1,BB2
3616
3617  012230  013700  012654                  MOV     BBGS,R0         ;FROM NOW ON FORCE
3618  012234  042700  177717                  BIC     #177717,R0      ;SELECT THE GROUP BEING
3619  012240  010037  177746                  MOV     R0,@#CONTRL     ;TESTED.
3620
3621  012244  012700  012630          BB3:    MOV     #BBADR1,R0      ;INITIALIZE.
3622  012250  012720  101442                  MOV     #BOTPRG,(R0)+   ;CONTAINS THE TEST ADDRESS.
3623  012254  005020                          CLR     (R0)+
3624  012256  005020                          CLR     (R0)+           ;CONTAINS THE LOGICAL 'OR'
3625  012260  005020                          CLR     (R0)+           ;OF FAILING ADDRESSES.
3626  012262  012720  177777                  MOV     #-1,(R0)+       ;CONTAINS THE LOGICAL 'AND'
3627  012266  012720  177777                  MOV     #-1,(R0)+       ;OF BAD ADDRESSES
3628
3629
3630  012272  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
3631  012276  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
3632  012302  012702  172300                  MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
3633  012306  012703  000010                  MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
3634  012312  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
3635  012314  077302                          SOB     R3,64$          ;ADDRESSES.
3636  012316  005020                          CLR     (R0)+
3637  012320  012720  000200                  MOV     #200,(R0)+
3638  012324  012720  000400                  MOV     #400,(R0)+
3639  012330  012720  000600                  MOV     #600,(R0)+
3640  012334  012720  001000                  MOV     #1000,(R0)+
3641  012340  012720  001200                  MOV     #1200,(R0)+
3642  012344  012720  001400                  MOV     #1400,(R0)+
3643  012350  012710  177600                  MOV     #177600,(R0)
3644
3645  012354  012737  000020  172516          MOV     #20,@#MMR3      ;TURN ON MEMORY MANAGEMENT.
3646  012362  012737  000001  177572          MOV     #1,@#MMR0
3647
3648  012370  005037  012646                  CLR     BBFLG2          ;INITIALIZE THE ERROR
3649  012374  005037  012650                  CLR     BBCNT1          ;FLAG AND COUNT.
3650  012400  005037  012652                  CLR     BBCNT1+2
3651
3652  012404  012737  012660  000114          MOV     #BBERR1,@#CACHVEC       ;PREPARE FOR ERRORS.
3653
3654  012412                          BB4:
3655
3656                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
3657  012412  023737  012144  012632          CMP     BBLOAD+2,BBADR1+2       ;COMPARE THE HIGH ORDER
3658  012420  001006                          BNE     64$             ;PARTS OF BBLOAD AND ARG2.
3659  012422  023737  012142  012630          CMP     BBLOAD,BBADR1   ;COMPARE THE LOW ORDER
3660  012430  001002                          BNE     64$             ;PARTS.
3661
3662
```

```
3663
3664  012432  000137  012450                 JMP     BB5             ;THEY WERE EQUAL!
3665
3666  012436  103402                 64$:    BLO     65$
3667  012440  000137  012546                 JMP     BB7             ;THE FIRST ADDRESS IS LARGER
3668                                                                  ;THAN THE SECOND!
3669  012444  000137  012450          65$:    JMP     BB5             ;THE FIRST IS LESS THAN THE
3670                                                                  ;SECOND.
3671
3672
3673  012450  012700  012630          BB5:    MOV     #BBADR1,R0      ;SET UP MEMORY MANAGEMENT.
3674
3675  012454  011003                          MOV     (R0),R3         ;GET THE PHYSICAL ADDRESS POINTED
3676  012456  042703  177700                  BIC     #177700,R3      ;TO BY R0 AND ESTABLISH
3677  012462  011005                          MOV     (R0),R5 ;A VIRTUAL ADDRESS WHICH
3678  012464  016004  000002                  MOV     2(R0),R4        ;WILL RELOCATE THROUGH
3679  012470  073427  177772                  ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
3680  012474  010537  172354                  MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
3681  012500  062703  140000                  ADD     #140000,R3      ;IN R3.
3682
3683
3684  012504  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
3685  012506  005713                          TST     (R3)            ;TRY TO GET A HIT.
3686  012510  005713                          TST     (R3)
3687
3688  012512  032737  000010  177752          BIT     #10,@#HITMIS    ;WAS IT A HIT?
3689  012520  001004                          BNE     BB6             ;BRANCH IF YES, OTHERWISE
3690                                                                  ;REPORT ERROR.
3691  012522  013737  012644  001636          MOV     BBFLG1,$TMP2
3692  012530  104036                  1$:     ERROR   36
3693
3694  012532  062737  000004  012630  BB6:    ADD     #4,BBADR1       ;MOVE TO NEXT WORD PAIR.
3695  012540  005537  012632                  ADC     BBADR1+2
3696  012544  000722                          BR      BB4
3697
3698  012546  005737  012646          BB7:    TST     BBFLG2          ;DID AN ERROR OCCUR IN
3699  012552  001410                          BEQ     BB8             ;THAT GROUP, IF YES PRINT
3700  012554  112737  000037  001514          MOVB    #37,$ITEMB      ;AN ERROR SUMMARY
3701  012562  013737  012644  001634          MOV     BBFLG1,$TMP1
3702  012570  004737  044334                  JSR     PC,ERTYPE
3703
3704  012574  005737  012644          BB8:    TST     BBFLG1          ;HAVE BOTH GROUPS BEEN TESTED?
3705  012600  001157                          BNE     BBDONE
3706  012602  012737  000001  012644          MOV     #1,BBFLG1       ;IF NOT, GO BACK AND
3707  012610  012737  000054  012654          MOV     #S1MOM1,BBGS    ;TEST GROUP 1
3708  012616  012737  000034  012656          MOV     #S0MOM1,BBGM
3709  012624  000137  012166                  JMP     BB1
3710
3711  012630  000000          BBADR1: .WORD   0               ;THE TEST ADDRESS.
3712  012632  000000                  .WORD   0
3713  012634  000000          BBADR2: .WORD   0               ;LOGICAL 'OR' OF BAD ADDRESSES.
3714  012636  000000                  .WORD   0
3715  012640  000000          BBADR3: .WORD   0               ;LOGICAL 'AND' OF BAD ADDRESSES.
3716  012642  000000                  .WORD   0
3717
3718  012644  000000          BBFLG1: .WORD   0               ;FLAG: 1, IF TESTING GROUP 1,
```

```
3719
3720  012646  000000                  BBFLG2: .WORD   0               ;OR 0, IF TESTING GROUP 0.
3721                                                                  ;ERROR FLAG: 0, IF NO ERRORS
3722                                                                  ;OCCURRED IN THE TESTED
3723  012650  000000                  BBCNT1: .WORD   0               ;GROUP.
3724  012652  000000                          .WORD   0               ;ERROR COUNT.
3725
3726  012654  000000                  BBGS:   .WORD   0               ;PATTERNS FOR THE CACHE
3727  012656  000000                  BBGM:   .WORD   0               ;CONTROL REGISTER
3728
3729  012660                          BBERR1:
3730
3731                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
3732  012660  023737  177742  012632          CMP     LOADRS+2,BBADR1+2       ;COMPARE THE HIGH ORDER
3733  012666  001006                          BNE     64$             ;PARTS OF LOADRS AND ARG2.
3734  012670  023737  177740  012630          CMP     LOADRS,BBADR1   ;COMPARE THE LOW ORDER
3735  012676  001002                          BNE     64$             ;PARTS.
3736
3737
3738
3739  012700  000137  012716                  JMP     BBERR2          ;THEY WERE EQUAL!
3740
3741  012704  103402                  64$:    BLO     65$
3742  012706  000137  043500                  JMP     SPUR            ;THE FIRST ADDRESS IS LARGER
3743                                                                  ;THAN THE SECOND!
3744  012712  000137  043500          65$:    JMP     SPUR            ;THE FIRST IS LESS THAN THE
3745                                                                  ;SECOND.
3746
3747
3748  012716  032737  000060  177744  BBERR2: BIT     #60,@#MEMERR    ;MAKE SURE A CACHE ADDRESS
3749  012724  001002                          BNE     BBERR3          ;MEMORY PARITY ERROR OCCURRED.
3750  012726  000137  043500                  JMP     SPUR
3751
3752  012732                          BBERR3:                         ;REPORT ERROR.
3753  012732  013737  012644  001640          MOV     BBFLG1,$TMP3
3754  012740  012637  001636                  MOV     (SP)+,$TMP2
3755  012744  005726                          TST     (SP)+
3756  012746  013737  177744  001642          MOV     @#MEMERR,$TMP4
3757  012754  013737  177740  001650          MOV     @#LOADRS,$TMP7
3758  012762  013737  177742  001652          MOV     @#HIADRS,$TMP10
3759  012770  013737  012630  001644          MOV     BBADR1,$TMP5
3760  012776  013737  012632  001646          MOV     BBADR1+2,$TMP6
3761  013004  104040                  1$:     ERROR   40
3762
3763  013006  053737  012630  012634          BIS     BBADR1,BBADR2   ;COMPUTE LOGICAL 'OR' OF
3764  013014  053737  012632  012636          BIS     BBADR1+2,BBADR2+2       ;BAD ADDRESSES.
3765  013022  005137  012640                  COM     BBADR3          ;COMPUT THE LOGICAL 'AND'
3766  013026  043737  012630  012640          BIC     BBADR1,BBADR3   ;OF THE BAD ADDRESSES.
3767  013034  005137  012640                  COM     BBADR3
3768  013040  005137  012642                  COM     BBADR3+2
3769  013044  043737  012632  012642          BIC     BBADR1+2,BBADR3+2
3770  013052  005137  012642                  COM     BBADR3+2
3771
3772  013056  012737  177777  012646          MOV     #-1,BBFLG2      ;SET THE ERROR FLAG.
3773  013064  005237  012650                  INC     BBCNT1          ;INCREMENT THE ERROR
3774  013070  005537  012652                  ADC     BBCNT1+2        ;COUNT.
```

```
3775
3776  013074  012737  013116  000114          MOV     #BBERR4,@#CACHVEC       ;TRY TO GET THE BAD
3777                                                                          ;ADDRESS OUT OF THE ADDRESS
3778                                                                          ;MEMORY.
3779  013102  013705  177740                  MOV     @#LOADRS,R5
3780  013106  042705  176001                  BIC     #176001,R5
3781  013112  005715                          TST     (R5)
3782  013114  000401                          BR      BBERR5
3783  013116  022626                  BBERR4: CMP     (SP)+,(SP)+
3784  013120  012737  177777  177744  BBERR5: MOV     #-1,@#MEMERR
3785  013126  012737  012660  000114          MOV     #BBERR1,@#CACHVEC
3786  013134  000137  012532                  JMP     BB6
3787
3788  013140  104407                  BBDONE: RSET                            ;DONE!
3789
3790                                  ;;*****************************************************************
3791                                  ;*TEST 11       CACHE ADDRESS MEMORY PARITY LOGIC TEST
3792                                  ;*
3793                                  ;*THIS IS A TEST OF THE PARITY CHECKERS AND PARITY GENERATOR
3794                                  ;*OF THE CACHE ADDRESS MEMORY. EVERY POSSIBLE ADDRESS TAG,
3795                                  ;*BITS 21 THROUGH 10, WHICH CAN BE STORED IN THE CACHE
3796                                  ;*ADDRESS MEMORY IS GENERATED, MADE A HIT AND THE
3797                                  ;*MAINTENANCE REGISTER IS THEN USED TO FORCE A CACHE ADDRESS
3798                                  ;*MEMORY PARITY ERROR AT EACH OF THE ADDRESSES
3799                                  ;*GENERATED. NOTE THAT BITS 9 THROUGH 0 OF THE ADDRESSES
3800                                  ;*IS NOT OF CONCERN, SO THESE BITS WILL BE THE SAME
3801                                  ;*FOR EACH ADDRESS; THIS IS BECAUSE ONLY BITS 21 THROUGH
3802                                  ;*10 ARE STORED IN THE ADDRESS MEMORY THEREFORE ONLY
3803                                  ;*THESE BITS ARE PARITY CHECKED IN THE CACHE ADDRESS
3804                                  ;*MEMORY PARITY CHECKERS. ALSO NOTE THAT THE RANGE
3805                                  ;*OF THE ADDRESSES MUST BE LIMITED TO BETWEEN THE
3806                                  ;*BOUNDS IMPOSED BY THE HIGHEST AVAILABLE MEMORY WORD
3807                                  ;*AND THE LAST WORD OF MEMORY USED BY THIS PROGRAM.
3808                                  ;*THE MANNER IN WHICH THE ERROR WILL BE FORCED
3809                                  ;*WILL BE TO PUT THE INSTRUCTIONS:
3810                                  ;*      1$:     MOV     R4,(R2)
3811                                  ;*      TSTADS: CLR     (R2)
3812                                  ;*              RTS     PC
3813                                  ;*AT THE PARTICULAR ADDRESS BEING TESTED, WHERE
3814                                  ;*'TSTADS' IS THE ADDRESS BEING TESTED. R4 CONTAINS
3815                                  ;*A PATTERN TO BE LOADED IN THE MAINTENANCE REGISTER
3816                                  ;*WHICH WILL FORCE AN ERROR IN THE CACHE ADDRESS
3817                                  ;*MEMORY; R2 CONTAINS THE ADDRESS OF THE MAINTENANCE
3818                                  ;*REGISTER. NOTE FOR EACH ADDRESS R4 WILL FIRST
3819                                  ;*BE SUCH AS TO CAUSE AN ERROR IN THE LOW
3820                                  ;*BYTE ADDRESS PARITY CHECKER THEN AT THE SAME
3821                                  ;*ADDRESS AN ERROR WILL BE FORCED ON THE HIGH BYTE!
3822                                  ;*THE SEQUENCE OF TEST ADDRESSES WILL BE GENERATED
3823                                  ;*TWICE ONCE MAKING THEM HITS IN GROUP 0 THEN
3824                                  ;*MAKING THEM HITS IN GROUP 1.
3825                                  ;*
3826                                  ;*****************************************************************
3827  013142  000004                  TST11:  SCOPE
3828  013144  012737  000002  001702          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
3829          000011                  AA=$TN-1
3830                                                                  ;SET THE SKAD REGISTER
```

```
3831  013152  012737  014232  043632          MOV     #TST12,SKAD     ;IN CASE THE TEST ABORTS.
3832
3833  013160  113737  001502  001632          MOVB    $TSTNM,$TMPO
3834  013166  012737  043500  000114          MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
3835  013174  104411                          MMSKIP
3836
3837  013176  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
3838  013202  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
3839  013206  012702  172300                  MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
3840  013212  012703  000010                  MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
3841  013216  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
3842  013220  077302                          SOB     R3,64$          ;ADDRESSES.
3843  013222  005020                          CLR     (R0)+
3844  013224  012720  000200                  MOV     #200,(R0)+
3845  013230  012720  000400                  MOV     #400,(R0)+
3846  013234  012720  000600                  MOV     #600,(R0)+
3847  013240  012720  001000                  MOV     #1000,(R0)+
3848  013244  012720  001200                  MOV     #1200,(R0)+
3849  013250  012720  001400                  MOV     #1400,(R0)+
3850  013254  012710  177600                  MOV     #177600,(R0)
3851
3852  013260  104412                          SIZE
3853  013262  000000                  AALOAD: .WORD   0               ;ADDRESS OF THE HIGHEST
3854  013264  000000                  AAHIAD: .WORD   0               ;WORD IN MEMORY.
3855  013266  042737  000002  013262          BIC     #2,AALOAD
3856
3857  013274  012700  014072                  MOV     #AATMP2,R0      ;ESTABLISH BITS 9 THROUGH
3858  013300  042700  176003                  BIC     #176003,R0      ;0 TO BE PART OF ALL
3859  013304  010037  014056                  MOV     R0,AAOFST       ;THE TEST ADDRESSES.
3860  013310  005037  014060                  CLR     AAOFST+2
3861
3862  013314  012737  000020  172516          MOV     #20,@#MMR3      ;ENABLE 22-BIT MODE
3863  013322  012737  000001  177572          MOV     #1,@#MMR0       ;ADDRESSING
3864
3865  013330  012737  000030  014046          MOV     #SOM1,AAGS      ;TEST GROUP 0 FIRST, AAGS
3866  013336  005037  014042                  CLR     AAFLG1          ;CONTAINS A PATTERN TO
3867  013342  012737  001400  014050          MOV     #1400,AAERGS    ;BE PUT IN THE CONTROL
3868  013350  012737  004420  014066          MOV     #4420,AAEXER    ;REGISTER. AAERGS CONTAINS
3869                                                                  ;A PATTERN FOR THE MAINT. REG.
3870  013356  012737  000001  014054  AA1:    MOV     #1,AAADR1+2     ;AAADR1 CONTAINS BITS
3871  013364  005037  014052                  CLR     AAADR1          ;10 THROUGH 22 OF
3872                                                                  ;THE TEST ADDRESS.
3873                                                                  ;INITIALIZE IT.
3874  013370  013737  014046  177746          MOV     AAGS,@#CONTRL   ;SELECT THE GROUP BEING
3875                                                                  ;TESTED. MISS THE OTHER
3876                                                                  ;GROUP.
3877  013376                          AA2:                            ;GET THE TEST ADDRESS
3878                                                                  ;INTO THE AAADR2=AAADR1+AAOFST
3879                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
3880  013376  013737  014052  014062          MOV     AAADR1,AAADR2
3881  013404  013737  014054  014064          MOV     AAADR1+2,AAADR2+2
3882  013412  063737  014056  014062          ADD     AAOFST,AAADR2
3883  013420  005537  014064                  ADC     AAADR2+2
3884  013424  063737  014060  014064          ADD     AAOFST+2,AAADR2+2
3885
3886
```

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

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
013432  023737  014064  013264        CMP     AAADR2+2,AALOAD+2       ;COMPARE THE HIGH ORDER
013440  001006                        BNE     64$                     ;PARTS OF AAADR2 AND ARG2.
013442  023737  014062  013262        CMP     AAADR2,APLOAD           ;COMPARE THE LOW ORDER
013450  001002                        BNE     64$                     ;PARTS.


013452  000137  013470                JMP     AA2                     ;THEY WERE EQUAL!

013456  103402                64$:    BLO     65$
013460  000137  014000                JMP     AA8                     ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
013464  000137  013470        65$:    JMP     AA3                     ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.


013470  012737  000001  014044  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.
013476  013703  014062                MOV     @AAADR2,R3
013502  013702  014064                MOV     AAADR2+2,R2
013506  162703  000002                SUB     #2,R3
013512  005602                        SBC     R2
013514  010300                        MOV     R3,R0
013516  042700  177700                BIC     #177700,R0              ;R0 CONTAINS THE VIRTUAL
013522  062700  140000                ADD     #140000,R0              ;ADDRESS.

013526  073227  177772                ASHC    #-6,R2                  ;SET KIPAR6
013532  010337  172354                MOV     R3,@#KIPAR6

013536  012737  043500  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
013544  012710  010112                MOV     #010112,(R0)            ;010112 = 'MOV R4,(R2)'
013550  012760  005012  000002        MOV     #005012,2(R0)           ;005012 = 'CLR (R2)'
013556  012760  000207  000004        MOV     #000207,4(R0)           ;000207 = 'RTS PC'

013564  005760  000002                TST     2(R0)                   ;MAKE THE TEST ADDRESS
013570  005760  000002                TST     2(R0)                   ;A HIT IN THE GROUP
013574  032737  000010  177752        BIT     #10,@#HITMIS            ;BEING TESTED!
013602  001016                        BNE     AA4
```

```
 3943  013604  012737  013632  001640          MOV     #1$,$TMP3       ;IF UNABLE TO GET A GIT
 3944  013612  013737  014042  001634          MOV     AAFLG1,$TMP1    ;REPORT ERROR!
 3945  013620  010037  001636                  MOV     R0,$TMP2
 3946  013624  062737  000002  001636          ADD     #2,$TMP2
 3947  013632  104001                  1$:     ERROR   1
 3948  013634  000137  013762                  JMP     AA7             ;GO TO NEXT TEST ADDRESS.
 3949
 3950  013640                          AA4:                            ;THE TEST ADDRESS IS NOW
 3951                                                                  ;A HIT IN THE GROUP
 3952  013640  012737  014076  000114          MOV     #AAERR1,@#CACHVEC       ;BEING TESTED. NOW RESET
 3953                                                                  ;CACHVEC TO GO TO THE EXPECTED
 3954                                                                  ;ERROR HANDLER
 3955  013646  012702  177750                  MOV     #MAINT,R2       ;SET R2
 3956  013652  013704  014050                  MOV     AAERGS,R4       ;SET R4 WHICH WILL BE
 3957  013656  042704  005000                  BIC     #5000,R4        ;LOADED INTO THE MAINT.
 3958                                                                  ;REG SO AS TO FORCE
 3959                                                                  ;A LOW BYTE ADDRESS
 3960                                                                  ;MEMORY PARITY ERROR
 3961                                                                  ;IN THE GROUP BEING
 3962                                                                  ;TESTED.
 3963  013662  000240                  NOP                             ;FOR SCOPING WITH AN OSCILLOSCOPE.
 3964  013664  004710                  JSR     PC,(R0)                 ;GO TO THE TEST
 3965                                                                  ;ADDRESS!
 3966
 3967  013666                          AA5:                            ;RETURN,RTS PC, BACK TO
 3968  013666  013737  014042  001636          MOV     AAFLG1,$TMP2    ;HERE IF THE TEST FAILED
 3969  013674  013737  014062  001640          MOV     AAADR2,$TMP3    ;TO FORCE AN ERROR AT
 3970  013702  013737  014064  001642          MOV     AAADR2+2,$TMP4  ;THE TEST ADDRESS'S LOW
 3971  013710  104136                  1$:     ERROR   136             ;BYTE. REPORT THE FAILURE!
 3972
 3973  013712                          AA6:                            ;TRY TO DO THE SAME
 3974                                                                  ;THING NOW ONLY FORCE THE
 3975                                                                  ;ERROR ON THE ADDRESSES
 3976                                                                  ;HIGH BYTE!
 3977                                                                  ;THE INSTRUCTIONS ARE
 3978                                                                  ;ALREADY AT THE TEST
 3979  013712  012737  000002  014044          MOV     #2,AAFLG2       ;ADDRESS. BUT MAKE SURE
 3980  013720  005760  000002                  TST     2(R0)           ;IT IS STILL A HIT!
 3981  013724  013704  014050                  MOV     AAERGS,R4       ;SET R4 TO FORCE THE
 3982  013730  042704  002400                  BIC     #2400,R4        ;ERROR ON THE HIGH BYTE.
 3983  013734  004710                  JSR     PC,(R0)                 ;GO DO THE TEST!
 3984
 3985  013736                          AA16:                           ;RETURN,RTS PC, BACK TO HERE
 3986  013736  013737  014042  001636          MOV     AAFLG1,$TMP2    ;IF THE TEST FAILED
 3987  013744  013737  014062  001640          MOV     AAADR2,$TMP3    ;IN TRYING TO FORCE A
 3988  013752  013737  014064  001642          MOV     AAADR2+2,$TMP4  ;ERROR ON THE HIGH BYTE
 3989  013760  104137                  1$:     ERROR   137             ;IN THE ADDRESS MEMORY
 3990
 3991  013762  062737  002000  014052  AA7:    ADD     #2000,AAADR1    ;INCREMENT BITS 21 THROUGH
 3992  013770  005537  014054                  ADC     AAADR1+2        ;10 OF THE TEST ADDRESS
 3993  013774  000137  013376                  JMP     AA2             ;AND GO TEST THIS NEW
 3994                                                                  ;TEST ADDRESS!
 3995  014000  005737  014042          AA8:    TST     AAFLG1          ;SEE IF BOTH GROUPS HAVE
 3996  014004  001111                  BNE     AADONE                  ;BEEN TESTED. IF NOT, GO
 3997  014006  012737  004440  014066          MOV     #4440,AAEXER    ;BACK TO AA1 TO TEST
 3998  014014  012737  000044  014046          MOV     #S1MO,AAGS      ;GROUP ONE, OTHERWISE DONE!
```

```
 3999  014022  012737  000001  014042          MOV     #1,AAFLG1
 4000  014030  012737  006000  014050          MOV     #6000,AAERGS
 4001  014036  000137  013356                  JMP     AA1
 4002
 4003  014042  000000                AAFLG1: .WORD   0               ;A FLAG WHICH INDICATES
 4004                                                                  ;WHICH GROUP IS BEING TESTED
 4005                                                                  ;1 OR 0
 4006  014044  000000                AAFLG2: .WORD   0               ;A FLAG WHICH INDICATES
 4007                                                                  ;WHETHER THE LOW BYTE (1)
 4008                                                                  ;THE HIGH BYTE (2) IS
 4009                                                                  ;BEING TESTED.
 4010  014046  000000                AAGS:   .WORD   0               ;A PATTERN FOR THE CONTROL
 4011                                                                  ;REGISTER.
 4012  014050  000000                AAERGS: .WORD   0               ;PATTERN FOR THE MAINT. REG.
 4013  014052  000000                AAADR1: .WORD   0               ;BITS 21 THROUGH 10 OF
 4014  014054  000000                        .WORD   0               ;THE TEST ADDRESS.
 4015  014056  000000                AAOFST: .WORD   0               ;BITS 9 THROUGH 0 OF
 4016  014060  000000                        .WORD   0               ;THE TEST ADDRESS.
 4017  014062  000000                AAADR2: .WORD   0               ;THE TEST ADDRESS
 4018  014064  000000                        .WORD   0               ;'AAADR2 = AAADR1+AAOFST'
 4019  014066  000000                AAEXER: .WORD   0               ;EXPECTED ERROR REGISTER
 4020  014070  000000                AATMP1: .WORD   0               ;THESE ADDRESSES ARE
 4021  014072  000000                AATMP2: .WORD   0               ;USED TO DETERMINE AAOFST.
 4022  014074  000000                        .WORD   0
 4023
 4024  014076  016666  000002  000004  AAERR1: MOV     2(SP),4(SP)     ;RESET THE STACK. RECALL THAT THE
 4025  014104  012616                          MOV     (SP)+,(SP)              ;TEST ROUTINE WAS JSR'ED TO AND
 4026                                                                  ;A PARITY ERROR TRAP BROUGHT CONTROL
 4027                                                                  ;BACK!!
 4028  014106  023737  014066  177744          CMP     AAEXER,@#MEMERR ;MAKE SURE THE ERROR
 4029  014114  001405                          BEQ     1$              ;WHICH OCCURRED WAS
 4030  014116  012737  043500  000114          MOV     #SPUR,@#CACHVEC ;THE EXPECTED ERROR AT
 4031  014124  000137  043500                  JMP     SPUR            ;THE EXPECTED ADDRESS,
 4032                                                                  ;IF NOT GO TO THE
 4033                                                                  ;SPURIOUS ERROR HANDLER,
 4034                                                                  ;SPUR!
 4035  014130                        1$:
 4036
 4037                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
 4038  014130  023737  014064  177742          CMP     AAADR2+2,LOADRS+2       ;COMPARE THE HIGH ORDER
 4039  014136  001006                          BNE     64$             ;PARTS OF AAADR2 AND ARG2.
 4040  014140  023737  014062  177740          CMP     AAADR2,LOADRS   ;COMPARE THE LOW ORDER
 4041  014146  001002                          BNE     64$             ;PARTS.
 4042
 4043
 4044
 4045  014150  000137  014166                  JMP     AAERR2          ;THEY WERE EQUAL!
 4046
 4047  014154  103402                64$:    BLO     65$
 4048  014156  000137  043500                  JMP     SPUR            ;THE FIRST ADDRESS IS LARGER
 4049                                                                  ;THAN THE SECOND!
 4050  014162  000137  043500        65$:    JMP     SPUR            ;THE FIRST IS LESS THAN THE
 4051                                                                  ;SECOND.
 4052
 4053
 4054  014166  012737  177777  177744  AAERR2: MOV     #-1,@#MEMERR    ;IF EVERYTHING WAS
```

```
                                                                ;CORRECT, CLR THE ERROR
014174  022626                          CMP     (SP)+,(SP)+     ;REGISTER RESET THE
                                                                ;STACK AND CONTINUE
014176  023727  014044  000002          CMP     AAFLG2,#2       ;TESTING
014204  001002                          BNE     1$
014206  000137  013762                  JMP     AA7             ;TEST THE NEXT ADDRESS
014212  023727  014044  000001  1$:     CMP     AAFLG2,#1
014220  001002                          BNE     2$
014222  000137  013712                  JMP     AA6             ;TEST THE HIGH BYTE OF THIS ADDRESS
014226  000000                  2$:     HALT                    ;???HOW DID WE GET HERE?

014230  104407                  AADONE: RSET                    ;DONE!

                        ;;*************************************************************
                        ;*TEST 12       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 WRITING 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.
                        ;*
                        ;;*************************************************************
014232  000004          TST12:  SCOPE
014234  012737  000002  001702          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
        000012          UU=$TN-1
014242                  UU0:
                                                                ;SET THE SKAD REGISTER
014242  012737  015654  043632          MOV     #TST13,SKAD     ;IN CASE THE TEST ABORTS.

014250  012737  043500  000114          MOV     #SPUR,@#CACHVEC ;AT FIRST EXPECT NO ERRORS
014256  113737  001502  001632          MOVB    $TSTNM,$TMP0
014264  005037  015342                  CLR     UUFLG3          ;ERROR FLAG.
014270  104411                          MMSKIP

014272  104412                          SIZE
014274  000000          UULOAD: .WORD   0               ;ADDRESS OF THE HIGHEST WORD
014276  000000          UUHIAD: .WORD   0               ;IN MEMORY

014300  005037  015336                  CLR     UUFLG1          ;TEST GROUP 0 FIRST.
014304  012737  000034  015360          MOV     #S0MOM1,UUGS
014312  012737  000054  015362          MOV     #S1MOM1,UUGM
```

```
4111
4112  014320  005037  015340          UU1:    CLR     UUFLG2          ;CLEAR THE PROGRESS FLAG.
4113  014324  012700  014242                  MOV     #UU3,R0         ;MAKE THIS CODE HITS, IN
4114  014330  012701  001000                  MOV     #1000,R1        ;THE GROUP NOT BEING TESTED.
4115
4116  014334  013737  015360  177746  UU2:    MOV     UUGS,@#CONTRL
4117  014342  005760  002000                  TST     2000(R0)
4118  014346  013737  015362  177746          MOV     UUGM,@#CONTRL
4119  014354  005720                          TST     (R0)+
4120  014356  077112                          SOB     R1,UU2
4121
4122  014360  013701  015360                  MOV     UUGS,R1         ;SELECT THE GROUP BEING TESTED.
4123  014364  042701  177717                  BIC     #177717,R1
4124  014370  010137  177746                  MOV     R1,@#CONTRL
4125
4126
4127  014374  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
4128  014400  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
4129  014404  012702  172300                  MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
4130  014410  012703  000010                  MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
4131  014414  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
4132  014416  077302                          SOB     R3,64$          ;ADDRESSES.
4133  014420  005020                          CLR     (R0)+
4134  014422  012720  000200                  MOV     #200,(R0)+
4135  014426  012720  000400                  MOV     #400,(R0)+
4136  014432  012720  000600                  MOV     #600,(R0)+
4137  014436  012720  001000                  MOV     #1000,(R0)+
4138  014442  012720  001200                  MOV     #1200,(R0)+
4139  014446  012720  001400                  MOV     #1400,(R0)+
4140  014452  012710  177600                  MOV     #177600,(R0)
4141
4142  014456  012737  000020  172516          MOV     #20,@#MMR3      ;TURN ON MEMORY MANAGEMENT.
4143  014464  012737  000001  177572          MOV     #1,@#MMR0
4144
4145  014472  005037  015350                  CLR     UUADR2          ;INITIALIZE THE ADDRESSES.
4146  014476  005037  015352                  CLR     UUADR2+2
4147  014502  012737  140000  015344          MOV     #140000,UUADR1
4148  014510  005037  015346                  CLR     UUADR1+2
4149  014514  012701  000400                  MOV     #400,R1         ;COUNTER.
4150  014520  012737  015366  000114          MOV     #UUERR1,@#CACHVEC
4151  014526  012737  000001  015340          MOV     #1,UUFLG2       ;KEEP TRACK OF TEST PROGRESS.
4152  014534                          UU3:
4153                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
4154  014534  013737  015344  015354          MOV     UUADR1,UUADR3
4155  014542  013737  015346  015356          MOV     UUADR1+2,UUADR3+2
4156  014550  063737  015350  015354          ADD     UUADR2,UUADR3
4157  014556  005537  015356                  ADC     UUADR3+2
4158  014562  063737  015352  015356          ADD     UUADR2+2,UUADR3+2
4159
4160
4161
4162
4163  014570                          UU4:
4164
4165                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
4166  014570  023737  015356  014276          CMP     UUADR3+2,UULOAD+2       ;COMPARE THE HIGH ORDER
```

```
4167  014576  001006                          BNE     64$             ;PARTS OF UUADR3 AND ARG2.
4168  014600  023737  015354  014274          CMP     UUADR3,UULOAD   ;COMPARE THE LOW ORDER
4169  014606  001002                          BNE     64$             ;PARTS.
4170
4171
4172
4173  014610  000137  014642                  JMP     UU6             ;THEY WERE EQUAL!
4174
4175  014614  103402                  64$:    BLO     65$
4176  014616  000137  014626                  JMP     UU5             ;THE FIRST ADDRESS IS LARGER
4177                                                                  ;THAN THE SECOND!
4178  014622  000137  014642          65$:    JMP     UU6             ;THE FIRST IS LESS THAN THE
4179                                                                  ;SECOND.
4180
4181
4182  014626  012737  140000  015344  UU5:    MOV     #140000,UUADR1  ;RESET TO GET VALID ADDRESS.
4183  014634  005037  015346                  CLR     UUADR1+2
4184  014640  000735                          BR      UU3
4185
4186  014642  012702  015354          UU6:    MOV     #UUADR3,R2
4187
4188  014646  011203                          MOV     (R2),R3         ;GET THE PHYSICAL ADDRESS POINTED
4189  014650  042703  177700                  BIC     #177700,R3      ;TO BY R2 AND ESTABLISH
4190  014654  011205                          MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
4191  014656  016204  000002                  MOV     2(R2),R4        ;WILL RELOCATE THROUGH
4192  014662  073427  177772                  ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
4193  014666  010537  172354                  MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
4194  014672  062703  140000                  ADD     #140000,R3      ;IN R3.
4195
4196
4197  014676  005713                          TST     (R3)            ;GET A HIT AT THE TEST
4198  014700  005713                          TST     (R3)            ;ADDRESS.
4199
4200  014702  032737  000010  177752          BIT     #10,@#HITMIS
4201  014710  001012                          BNE     UU7
4202  014712  013737  015336  001636          MOV     UUFLG1,$TMP2
4203  014720  013737  015354  001640          MOV     UUADR3,$TMP3
4204  014726  013737  015356  001642          MOV     UUADR3+2,$TMP4
4205  014734  104041                  1$:     ERROR   41
4206  014736  062737  002000  015344  UU7:    ADD     #2000,UUADR1
4207  014744  005537  015346                  ADC     UUADR1+2
4208  014750  062737  000004  015350          ADD     #4,UUADR2       ;LOOP TO WRITE NEXT ADDRESS
4209  014756  005301                          DEC     R1
4210  014760  001402                          BEQ     1$
4211  014762  000137  014534                  JMP     UU3
4212  014766  012737  000002  015340  1$:     MOV     #2,UUFLG2
4213
4214  014774  013700  015362                  MOV     UUGM,R0         ;FROM NOW ON SELECT THE
4215  015000  042700  177717                  BIC     #177717,R0      ;GROUP NOT BEING TESTED.
4216  015004  010037  177746                  MOV     R0,@#CONTRL
4217
4218  015010  005037  015350          UU8:    CLR     UUADR2          ;NOW RE-GENERATE ALL THE
4219  015014  005037  015352                  CLR     UUADR2+2        ;ADDRESS WHICH WERE MADE
4220  015020  012737  140000  015344          MOV     #140000,UUADR1  ;HITS, ABOVE, AND MAKE SURE
4221  015026  005037  015346                  CLR     UUADR1+2        ;THEY ARE STILL HITS.
4222  015032  012701  000400                  MOV     #400,R1
```

```
4223  015036  012737  000003  015340          MOV     #3,UUFLG2
4224  015044
4225                                  UU9:
4226                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
4226  015044  013737  015344  015354          MOV     UUADR1,UUADR3
4227  015052  013737  015346  015356          MOV     UUADR1+2,UUADR3+2
4228  015060  063737  015350  015354          ADD     UUADR2,UUADR3
4229  015066  005537  015356                  ADC     UUADR3+2
4230  015072  063737  015352  015356          ADD     UUADR2+2,UUADR3+2
4231
4232
4233
4234
4235  015100                                  UU10:
4236
4237                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
4238  015100  023737  015356  014276          CMP     UUADR3+2,UULOAD+2       ;COMPARE THE HIGH ORDER
4239  015106  001006                          BNE     64$             ;PARTS OF UUADR3 AND ARG2.
4240  015110  023737  015354  014274          CMP     UUADR3,UULOAD   ;COMPARE THE LOW ORDER
4241  015116  001002                          BNE     64$             ;PARTS.
4242
4243
4244
4245  015120  000137  015152                  JMP     UU12            ;THEY WERE EQUAL!
4246
4247  015124  103402                  64$:    BLO     65$
4248  015126  000137  015136                  JMP     UU11            ;THE FIRST ADDRESS IS LARGER
4249                                                                  ;THAN THE SECOND!
4250  015132  000137  015152          65$:    JMP     UU12            ;THE FIRST IS LESS THAN THE
4251                                                                  ;SECOND.
4252
4253
4254  015136  012737  140000  015344  UU11:   MOV     #140000,UUADR1  ;RESET TO GET A VALID ADDRESS.
4255  015144  005037  015346                  CLR     UUADR1+2
4256  015150  000735                          BR      UU9
4257
4258  015152  012702  015354          UU12:   MOV     #UUADR3,R2
4259
4260  015156  011203                          MOV     (R2),R3         ;GET THE PHYSICAL ADDRESS POINTED
4261  015160  042703  177700                  BIC     #177700,R3      ;TO BY R2 AND ESTABLISH
4262  015164  011205                          MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
4263  015166  016204  000002                  MOV     2(R2),R4        ;WILL RELOCATE THROUGH
4264  015172  073427  177772                  ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
4265  015176  010537  172354                  MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
4266  015202  062703  140000                  ADD     #140000,R3      ;IN R3.
4267
4268
4269  015206  005713                          TST     (R3)    ;STILL A HIT?
4270  015210  032737  000010  177752          BIT     #10,@#HITMIS
4271  015216  001012                          BNE     UU13
4272                                                                  ;NOT A HIT, A DUAL ADDRESSING
4273  015220  013737  015336  001636          MOV     UUFLG1,$TMP2    ;PROBLEM?
4274  015226  013737  015354  001640          MOV     UUADR3,$TMP3
4275  015234  013737  015356  001642          MOV     UUADR3+2,$TMP4
4276  015242  104042                  1$:     ERROR   42
4277
4278  015244  062737  002000  015344  UU13:   ADD     #2000,UUADR1
```

```
4279  015252  005537  015346                    ADC     UUADR1+2
4280  015256  062737  000004  015350            ADD     #4,UUADR2       ;LOOP TO READ NEXT ADDRESS
4281  015264  005301                            DEC     R1
4282  015266  001402                            BEQ     1$
4283  015270  000137  015044                    JMP     UU9
4284  015274  012737  000004  015340  1$:       MOV     #4,UUFLG2
4285  015302  005737  015336          UU14:     TST     UUFLG1  ;TESTED BOTH GROUPS?
4286  015306  001161                            BNE     UUDONE          ;YES.
4287  015310  012737  000001  015336            MOV     #1,UUFLG1       ;NO, GO TEST GROUP 1.
4288  015316  012737  000054  015360            MOV     #S1MOM1,UUGS
4289  015324  012737  000034  015362            MOV     #SOMOM1,UUGM
4290  015332  000137  014320                    JMP     UU1
4291
4292  015336  000000                  UUFLG1:   .WORD   0               ;FLAG WHICH DESIGNATES
4293                                                                    ;WHICH GROUP IS BEING TESTED,
4294                                                                    ;1 OR 0.
4295  015340  000000                  UUFLG2:   .WORD   0               ;DESIGNATES HOW FAR THE
4296                                                                    ;TEST HAS PROGRESSED.
4297  015342  000000                  UUFLG3:   .WORD   0               ;ERROR DURING TEST UUFLG2=4
4298                                                                    ;PHASE.
4299  015344  000000                  UUADR1:   .WORD   0               ;ADDRESS WRITTEN INTO ADDRESS
4300  015346  000000                            .WORD   0               ;MEMORY LOCATION
4301  015350  000000                  UUADR2:   .WORD   0               ;ADDRESS MEMORY LOCATION
4302  015352  000000                            .WORD   0               ;BEING TESTED
4303  015354  000000                  UUADR3:   .WORD   0               ;TEST ADDRESS:UUADR3=UUADR1+UUADR2
4304  015356  000000                            .WORD   0
4305
4306  015360  000000                  UUGS:     .WORD   0               ;PATTERNS FOR THE CACHE CONTROL
4307  015362  000000                  UUGM:     .WORD   0               ;REGISTER.
4308  015364  000000                  UUTMP:    .WORD   0
4309
4310  015366  032737  000060  177744  UUERR1:   BIT     #60,@#MEMERR    ;WAS THE ERROR A CACHE ADDRESS
4311  015374  001002                            BNE     UUERR2          ;MEMORY PARITY ERROR
4312  015376  000137  043500                    JMP     SPUR
4313
4314  015402                          UUERR2:                           ;REPORT ERROR.
4315  015402  012637  001636                    MOV     (SP)+,$TMP2
4316  015406  005726                            TST     (SP)+
4317  015410  013737  015336  001640            MOV     UUFLG1,$TMP3
4318  015416  013737  177744  001642            MOV     @#MEMERR,$TMP4
4319  015424  013737  015354  001644            MOV     UUADR3,$TMP5
4320  015432  013737  015356  001646            MOV     UUADR3+2,$TMP6
4321  015440  013737  177740  001650            MOV     @#LOADRS,$TMP7
4322  015446  013737  177742  001652            MOV     @#HIADRS,$TMP10
4323  015454  104043                  1$:       ERROR   43
4324
4325  015456  042737  177717  001642            BIC     #177717,$TMP4   ;TRY TO GET THE BAD ADDRESS
4326  015464  013737  177746  015364            MOV     @#CONTRL,UUTMP  ;OUT OF THE ADDRESS MEMORY.
4327  015472  012737  015522  000114            MOV     #UUERR3,@#CACHVEC
4328  015500  013705  177740                    MOV     @#LOADRS,R5
4329  015504  042705  176001                    BIC     #176001,R5
4330  015510  013737  001642  177746            MOV     $TMP4,@#CONTRL
4331  015516  005715                            TST     (R5)
4332  015520  000401                            BR      UUERR4
4333  015522  022626                  UUERR3:   CMP     (SP)+,(SP)+
4334  015524  012737  177777  177744  UUERR4:   MOV     #-1,@#MEMERR
```

```
4335
4336  015532  013737  015364  177746          MOV     UUTMP,@#CONTRL  ;RESET THE CONTROL REGISTER.
4337  015540  012737  015366  000114          MOV     #UUERR1,@#CACHVEC
4338
4339  015546  023727  015340  000001          CMP     UUFLG2,#1       ;RETURN, USING UUFLG2 TO
4340  015554  001002                          BNE     1$              ;DECIDE WHERE.
4341  015556  000137  014736                  JMP     UU7
4342  015562  023727  015340  000002  1$:     CMP     UUFLG2,#2
4343  015570  001002                          BNE     2$
4344  015572  000137  015010                  JMP     UU8
4345  015576  023727  015340  000003  2$:     CMP     UUFLG2,#3
4346  015604  001002                          BNE     3$
4347  015606  000137  015244                  JMP     UU13
4348  015612  023727  015340  000004  3$:     CMP     UUFLG2,#4
4349  015620  001007                          BNE     4$
4350  015622  005737  015342                  TST     UUFLG3
4351  015626  001011                          BNE     UUDONE
4352  015630  005337  015342                  DEC     UUFLG3
4353  015634  000137  015302                  JMP     UU14
4354
4355  015640  005737  015340          4$:     TST     UUFLG2
4356  015644  001002                          BNE     UUDONE          ;??HALT???
4357  015646  000137  014320                  JMP     UU1
4358
4359  015652  104407                  UUDONE: RSET                    ;DONE!
4360
4361
4362                                  ;;*****************************************************************
4363                                  ;*TEST 13       CACHE ADDRESS MEMORY DUAL ADDRESS TEST, DOWNWARD
4364                                  ;*
4365                                  ;*THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS
4366                                  ;*MEMORY.  AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS'
4367                                  ;*ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP
4368                                  ;*BEING TESTED.  OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS
4369                                  ;*TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY
4370                                  ;*ON THE SYSTEM.  NOTE THAT HERE THE WORD 'TAG' REFERS
4371                                  ;*TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21,
4372                                  ;*WHICH ARE STORED IN THE CACHE ADDRESS MEMORY.  HERE
4373                                  ;*THE ADDRESS MEMORY IS WRITTEN IN THE DOWNWARD DIRECTION,
4374                                  ;*THAT IS 'TAG' 256 IS WRITTEN FIRST, 'TAG' 255 SECOND ETC.
4375                                  ;*THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED
4376                                  ;* TO SEE IF IT IS A HIT, THUS MAKING SURE NO
4377                                  ;*'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER
4378                                  ;*'TAG'.  NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL
4379                                  ;*ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD
4380                                  ;*INVOLVE WRITTING THE 'TAGS' IN THE UPWARD DIRECTION
4381                                  ;*AS WELL AS THE DOWNWARD DIRECTION.  THE UPWARD
4382                                  ;*WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND
4383                                  ;*IN TST12.
4384                                  ;*
4385                                  ;*****************************************************************
4386  015654  000004                  TST13:  SCOPE
4387  015656  012737  000002  001702          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
4388           000013                 VV=$TN-1
4389  015664                          VV0:
4390                                                                  ;SET THE SKAD REGISTER
```

```
4391  015664  012737  017302  043632          MOV     #TST14,SKAD     ;IN CASE THE TEST ABORTS.
4392
4393  015672  012737  043500  000114          MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
4394  015700  113737  001502  001632          MOVB    $TSTNM,$TMP0
4395
4396  015706  005037  016770                  CLR     VVFLG3          ;CLEAR THE ERROR FLAG.
4397
4398  015712  104411                          MMSKIP
4399
4400  015714  104412                          SIZE
4401  015716  000000                  VVLOAD: .WORD   0               ;ADDRESS OF THE HIGHEST
4402  015720  000000                  VVHIAD: .WORD   0               ;WORD IN MEMORY.
4403
4404  015722  005037  016764                  CLR     VVFLG1          ;TEST GROUP 0 FIRST
4405  015726  012737  000034  017006          MOV     #SCMOM1,VVGS
4406  015734  012737  000054  017010          MOV     #SIMOM1,VVGM
4407
4408  015742  005037  016766          VV1:    CLR     VVFLG2          ;CLEAR THE PROGRESS FLAG
4409  015746  012700  015664                  MOV     #VV0,R0         ;MAKE THIS CODE HITS IN
4410  015752  012701  001000                  MOV     #1000,R1        ;THE GROUP NOT BEING
4411                                                                  ;TESTED.
4412  015756  013737  017006  177746  VV2:    MOV     VVGS,@#CONTRL
4413  015764  005760  002000                  TST     2000(R0)
4414  015770  013737  017010  177746          MOV     VVGM,@#CONTRL
4415  015776  005720                          TST     (R0)+
4416  016000  077112                          SOB     R1,VV2
4417
4418  016002  013700  017006                  MOV     VVGS,R0         ;FROM NOW ON SELECT
4419  016006  042700  177717                  BIC     #177717,R0      ;THE GROUP BEING TESTED.
4420  016012  010037  177746                  MOV     R0,@#CONTRL
4421
4422
4423  016016  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
4424  016022  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
4425  016026  012702  172300                  MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
4426  016032  012703  000010                  MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
4427  016036  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
4428  016040  077302                          SOB     R3,64$          ;ADDRESSES.
4429  016042  005020                          CLR     (R0)+
4430  016044  012720  000200                  MOV     #200,(R0)+
4431  016050  012720  000400                  MOV     #400,(R0)+
4432  016054  012720  000600                  MOV     #600,(R0)+
4433  016060  012720  001000                  MOV     #1000,(R0)+
4434  016064  012720  001200                  MOV     #1200,(R0)+
4435  016070  012720  001400                  MOV     #1400,(R0)+
4436  016074  012710  177600                  MOV     #177600,(R0)
4437
4438  016100  012737  000020  172516          MOV     #20,@#MMR3      ;TURN ON MEMORY MANAGEMENT.
4439  016106  012737  000001  177572          MOV     #1,@#MMR0
4440
4441  016114  012737  001774  016776          MOV     #1774,VVADR2    ;INITIALIZE THE ADDRESSES
4442  016122  005037  017000                  CLR     VVADR2+2
4443  016126  012737  140000  016772          MOV     #140000,VVADR1
4444  016134  005037  016774                  CLR     VVADR1+2
4445  016140  012701  000400                  MOV     #400,R1         ;A COUNTER.
4446  016144  012737  017014  000114          MOV     #VVERR1,@#CACHVEC       ;EXPECT ERRORS NOW.
```

```
4447  016152  012737  000001  016766          MOV     #1,VVFLG2       ;KEEP TRACK OF TEST PROGRESS.
4448
4449  016160                          VV3:
4450                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
4451  016160  013737  016772  017002          MOV     VVADR1,VVADR3
4452  016166  013737  016774  017004          MOV     VVADR1+2,VVADR3+2
4453  016174  063737  016776  017002          ADD     VVADR2,VVADR3
4454  016202  005537  017004                  ADC     VVADR3+2
4455  016206  063737  017000  017004          ADD     VVADR2+2,VVADR3+2
4456
4457
4458
4459
4460  016214                          VV4:
4461
4462                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
4463  016214  023737  017004  015720          CMP     VVADR3+2,VVLOAD+2       ;COMPARE THE HIGH ORDER
4464  016222  001006                          BNE     64$             ;PARTS OF VVADR3 AND ARG2.
4465  016224  023737  017002  015716          CMP     VVADR3,VVLOAD   ;COMPARE THE LOW ORDER
4466  016232  001002                          BNE     64$             ;PARTS.
4467
4468
4469
4470  016234  000137  016266                  JMP     VV6             ;THEY WERE EQUAL!
4471
4472  016240  103402                  64$:    BLO     65$
4473  016242  000137  016252                  JMP     VV5             ;THE FIRST ADDRESS IS LARGER
4474                                                                  ;THAN THE SECOND!
4475  016246  000137  016266          65$:    JMP     VV6             ;THE FIRST IS LESS THAN THE
4476                                                                  ;SECOND.
4477
4478
4479  016252  012737  140000  016772  VV5:    MOV     #140000,VVADR1  ;RESET TO GET A VALID ADDRESS.
4480  016260  005037  016774                  CLR     VVADR1+2
4481  016264  000735                          BR      VV3
4482
4483  016266  012702  017002          VV6:    MOV     #VVADR3,R2
4484
4485  016272  011203                          MOV     (R2),R3         ;GET THE PHYSICAL ADDRESS POINTED
4486  016274  042703  177700                  BIC     #177700,R3      ;TO BY R2 AND ESTABLISH
4487  016300  011205                          MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
4488  016302  016204  000002                  MOV     2(R2),R4        ;WILL RELOCATE THROUGH
4489  016306  073427  177772                  ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
4490  016312  010537  172354                  MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
4491  016316  062703  140000                  ADD     #140000,R3      ;IN R3.
4492
4493
4494  016322  005713                          TST     (R3)            ;GET A HIT AT THE
4495  016324  005713                          TST     (R3)            ;TEST ADDRESS
4496  016326  032737  000010  177752          BIT     #10,@#HITMIS
4497  016334  001012                          BNE     VV7
4498                                                                  ;REPORT FAILURE TO GET A HIT.
4499  016336  013737  016764  001636          MOV     VVFLG1,$TMP2
4500  016344  013737  017002  001640          MOV     VVADR3,$TMP3
4501  016352  013737  017004  001642          MOV     VVADR3+2,$TMP4
4502  016360  104041                  1$:     ERROR   41
```

```
4503
4504  016362  062737  002000  016772  VV7:    ADD     #2000,VVADR1
4505  016370  005537  016774          ADC     VVADR1+2
4506  016374  062737  177774  016776          ADD     #-4,VVADR2      ;LOOP TO WRITE NEXT ADDRESS
4507  016402  005301                  DEC     R1
4508  016404  001402                  BEQ     1$
4509  016406  000137  016160          JMP     VV3
4510  016412  012737  000002  016766  1$:     MOV     #2,VVFLG2
4511
4512  016420  013700  017010          MOV     VVGM,R0         ;FROM NOW ON SELECT
4513  016424  042700  177717          BIC     #177717,R0      ;THE GROUP NOT BEING
4514  016430  010037  177746          MOV     R0,@#CONTRL     ;TESTED.
4515
4516  016434  012737  001774  016776  VV8:    MOV     #1774,VVADR2    ;NOW RE-GENERATE ALL THE
4517  016442  005037  017000          CLR     VVADR2+2        ;ADDRESSES MADE HITS IN
4518  016446  012737  140000  016772          MOV     #140000,VVADR1  ;THE ABOVE PORTION OF
4519  016454  005037  016774          CLR     VVADR1+2        ;THE TEST, AND MAKE SURE
4520  016460  012701  000400          MOV     #400,R1         ;THEY ARE STILL HITS.
4521  016464  012737  000003  016766          MOV     #3,VVFLG2
4522  016472                  VV9:
4523                  ;DOUBLE PRECISION ADDITION, UNSIGNED
4524  016472  013737  016772  017002          MOV     VVADR1,VVADR3
4525  016500  013737  016774  017004          MOV     VVADR1+2,VVADR3+2
4526  016506  063737  016776  017002          ADD     VVADR2,VVADR3
4527  016514  005537  017004          ADC     VVADR3+2
4528  016520  063737  017000  017004          ADD     VVADR2+2,VVADR3+2
4529
4530
4531
4532
4533  016526                  VV10:
4534
4535                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
4536  016526  023737  017004  015720          CMP     VVADR3+2,VVLOAD+2       ;COMPARE THE HIGH ORDER
4537  016534  001006                  BNE     64$             ;PARTS OF VVADR3 AND ARG2.
4538  016536  023737  017002  015716          CMP     VVADR3,VVLOAD   ;COMPARE THE LOW ORDER
4539  016544  001002                  BNE     64$             ;PARTS.
4540
4541
4542
4543  016546  000137  016600          JMP     VV12            ;THEY WERE EQUAL!
4544
4545  016552  103402                  BLO     65$
4546  016554  000137  016564          JMP     VV11            ;THE FIRST ADDRESS IS LARGER
4547                                                  ;THAN THE SECOND!
4548  016560  000137  016600  65$:    JMP     VV12            ;THE FIRST IS LESS THAN THE
4549                                                  ;SECOND.
4550
4551
4552  016564  012737  140000  016772  VV11:   MOV     #140000,VVADR1  ;RESET TO CREATE A VALID
4553  016572  005037  016774          CLR     VVADR1+2        ;ADDRESS
4554  016576  000735                  BR      VV9
4555
4556  016600  012702  017002  VV12:   MOV     #VVADR3,R2
4557
4558  016604  011203                  MOV     (R2),R3         ;GET THE PHYSICAL ADDRESS POINTED
```

```
4559  016606  042703  177700                  BIC     #177700,R3      ;TO BY R2 AND ESTABLISH
4560  016612  011205                          MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
4561  016614  016204  000002                  MOV     2(R2),R4        ;WILL RELOCATE THROUGH
4562  016620  073427  177772                  ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
4563  016624  010537  172354                  MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
4564  016630  062703  140000                  ADD     #140000,R3      ;IN R3.
4565
4566
4567  016634  005713                          TST     (R3)            ;STILL A HIT?
4568  016636  032737  000010  177752          BIT     #10,@#HITMIS
4569  016644  001012                          BNE     VV13
4570                                                                  ;REPORT ERROR.
4571  016646  013737  016764  001636          MOV     VVFLG1,$TMP2
4572  016654  013737  017002  001640          MOV     VVADR3,$TMP3
4573  016662  013737  017004  001642          MOV     VVADR3+2,$TMP4
4574  016670  104042                  1$:     ERROR   42
4575
4576  016672  062737  002000  016772  VV13:   ADD     #2000,VVADR1
4577  016700  005537  016774                  ADC     VVADR1+2
4578  016704  062737  177774  016776          ADD     #-4,VVADR2
4579  016712  005301                          DEC     R1
4580  016714  001402                          BEQ     1$
4581  016716  000137  016472                  JMP     VV9
4582  016722  012737  000004  016766  1$:     MOV     #4,VVFLG2
4583  016730  005737  016764          VV14:   TST     VVFLG1          ;TESTED BOTH GROUPS?
4584  016734  001161                          BNE     VVDONE          ;YES.
4585  016736  012737  000034  017010          MOV     #SOMOM1,VVGM    ;NO GO TEST GROUP 1.
4586  016744  012737  000054  017006          MOV     #S1MOM1,VVGS
4587  016752  012737  000001  016764          MOV     #1,VVFLG1
4588  016760  000137  015742                  JMP     VV1
4589
4590  016764  000000          VVFLG1: .WORD   0               ;0 OR 1, GROUP BEING TESTED.
4591  016766  000000          VVFLG2: .WORD   0               ;TEST PROGRESS FLAG.
4592  016770  000000          VVFLG3: .WORD   0               ;ERROR FLAG.
4593
4594  016772  000000          VVADR1: .WORD   0               ;PATTERN WRITTEN INTO THE ADDRESS
4595  016774  000000                  .WORD   0               ;MEMORY LOCATION.
4596  016776  000000          VVADR2: .WORD   0               ;ADDRESS MEMORY LOCATION BEING
4597  017000  000000                  .WORD   0               ;TESTED X 4.
4598  017002  000000          VVADR3: .WORD   0               ;TEST ADDRESS.
4599  017004  000000                  .WORD   0               ;VVADR3=VVADR2+VVADR1
4600
4601  017006  000000          VVGS:   .WORD   0               ;PATTERNS FOR THE CACHE
4602  017010  000000          VVGM:   .WORD   0               ;CONTROL REGISTER.
4603
4604  017012  000000          VVTMP:  .WORD   0
4605
4606  017014  032737  000060  177744  VVERR1; BIT     #60,@#MEMERR    ;WAS THE ERROR THAT CAUSED
4607  017022  001002                          BNE     VVERR2          ;THE TRAP TO HERE A CACHE
4608  017024  000137  043500                  JMP     SPUR            ;ADDRESS MEMORY PARITY ERROR?
4609
4610  017030                          VVERR2:                         ;REPORT ERROR.
4611  017030  012637  001636                  MOV     (SP)+,$TMP2
4612  017034  005726                          TST     (SP)+
4613  017036  013737  016764  001640          MOV     VVFLG1,$TMP3
4614  017044  013737  177744  001642          MOV     @#MEMERR,$TMP4
```

```
4615  017052  013737  017002  001644          MOV     VVADR3,$TMP5
4616  017060  013737  017004  001646          MOV     VVADR3+2,$TMP6
4617  017066  013737  177740  001650          MOV     @#LOADRS,$TMP7
4618  017074  013737  177742  001652          MOV     @#HIADRS,$TMP10
4619  017102  104043                  1$:     ERROR   43
4620
4621  017104  042737  177717  001642          BIC     #177717,$TMP4   ;TRY TO GET THE BAD ADDRESS
4622  017112  013737  177746  017012          MOV     @#CONTRL,VVTMP  ;OUT OF THE ADDRESS MEMORY.
4623  017120  012737  017150  000114          MOV     #VVERR3,@#CACHVEC
4624  017126  013705  177740                  MOV     @#LOADRS,R5
4625  017132  042705  176001                  BIC     #176001,R5
4626  017136  013737  001642  177746          MOV     $TMP4,@#CONTRL
4627  017144  005715                          TST     (R5)
4628  017146  000401                          BR      VVERR4
4629  017150  022626                  VVERR3: CMP     (SP)+,(SP)+
4630  017152  012737  177777  177744  VVERR4: MOV     #-1,@#MEMERR
4631
4632  017160  013737  017012  177746          MOV     VVTMP,@#CONTRL  ;RESET THE CONTRL REGISTER
4633  017166  012737  017014  000114          MOV     #VVERR1,@#CACHVEC
4634  017174  023727  016766  000001          CMP     VVFLG2,#1       ;RETURN, USING VVFLG2 TO
4635  017202  001002                          BNE     1$              ;DECIDE WHERE.
4636  017204  000137  016362                  JMP     VV7
4637  017210  023727  016766  000002  1$:     CMP     VVFLG2,#2
4638  017216  001002                          BNE     2$
4639  017220  000137  016434                  JMP     VV8
4640  017224  023727  016766  000003  2$:     CMP     VVFLG2,#3
4641  017232  001002                          BNE     3$
4642  017234  000137  016672                  JMP     VV13
4643  017240  023727  016766  000004  3$:     CMP     VVFLG2,#4
4644  017246  001007                          BNE     4$
4645  017250  005737  016770                  TST     VVFLG3
4646  017254  001011                          BNE     VVDONE
4647  017256  005337  016770                  DEC     VVFLG3
4648  017262  000137  016730                  JMP     VV14
4649  017266  005737  016766          4$:     TST     VVFLG2
4650  017272  001002                          BNE     VVDONE          ;????HALT???
4651  017274  000137  015742                  JMP     VV1
4652
4653  017300  104407                  VVDONE: RSET                    ;DONE!
4654
4655
4656                                  ;;*****************************************************************
4657                                  ;*TEST 14       CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ONES TEST
4658                                  ;*
4659                                  ;*THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC.  THIS
4660                                  ;*IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING
4661                                  ;*A WRITE.  IT DESIGNATES WHICH BYTES OF THE TWO WORDS OF
4662                                  ;*DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO
4663                                  ;*BE WRITTEN.  THIS WILL BE A TEST DOING CPU DATOB REFERENCES TO
4664                                  ;*THE CACHE.  THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN
4665                                  ;*OF ZEROES.
4666                                  ;*
4667                                  ;;*****************************************************************
4668  017302  000004                  TST14:  SCOPE
4669  017304  012737  000010  001702          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
4670          000014                  CC=$TN-1
```

```
                                                                ;SET THE SKAD REGISTER
017312  012737  020066  043632        MOV     #TST15,SKAD       ;IN CASE THE TEST ABORTS.

017320  113737  001502  001632        MOVB    $TSTNM,$TMPO
017326  012737  017572  000114        MOV     #CCERR1,@#CACHVEC

017334  012737  000014  177746        MOV     #MOM1,@#CONTRL    ;FORCE MISSES

017342  012700  017566                MOV     #CCTMP2,R0        ;LOCATE THE TEST SPACE.
017346  042700  000003                BIC     #3,R0
017352  010001                        MOV     R0,R1
017354  005010              CC1:      CLR     (R0)              ;TEST MASK 0
017356  005060  000002                CLR     2(R0)
017362  000240                        NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
017364  112711  000377                MOVB    #377,(R1)
017370  022710  000377                CMP     #377,(R0)
017374  001403                        BEQ     CC3
017376  004737  020004      CC2:      JSR     PC,CCERR3
017402  000403                        BR      CC4
017404  005760  000002      CC3:      TST     2(R0)
017410  001372                        BNE     CC2
017412  062701  000001      CC4:      ADD     #1,R1             ;TEST MASK 1.
017416  005010                        CLR     (R0)
017420  005060  000002                CLR     2(R0)
017424  000240                        NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
017426  112711  000377                MOVB    #377,(R1)
017432  022710  177400                CMP     #177400,(R0)
017436  001403                        BEQ     CC6
017440  004737  020004      CC5:      JSR     PC,CCERR3
017444  000403                        BR      CC7
017446  005760  000002      CC6:      TST     2(R0)
017452  001372                        BNE     CC5

017454  062701  000001      CC7:      ADD     #1,R1             ;TEST MASK 2.
017460  005010                        CLR     (R0)
017462  005060  000002                CLR     2(R0)
017466  000240                        NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
017470  112711  000377                MOVB    #377,(R1)
017474  022760  000377  000002        CMP     #377,2(R0)
017502  001403                        BEQ     CC9
017504  004737  020004      CC8:      JSR     PC,CCERR3
017510  000402                        BR      CC10
017512  005710              CC9:      TST     (R0)
017514  001373                        BNE     CC8

017516  062701  000001      CC10:     ADD     #1,R1             ;TEST MASK 3.
017522  005010                        CLR     (R0)
017524  005060  000002                CLR     2(R0)
017530  000240                        NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
017532  112711  000377                MOVB    #377,(R1)
017536  022760  177400  000002        CMP     #177400,2(R0)
017544  001403                        BEQ     CC12
017546  004737  020004      CC11:     JSR     PC,CCERR3
017552  000402                        BR      CC13
017554  005710              CC12:     TST     (R0)
017556  001373                        BNE     CC11
```

```
 4727
 4728  017560  000137  020064          CC13:   JMP     CCDONE
 4729
 4730  017564  000000                  CCTMP1: .WORD   0
 4731  017566  000000                  CCTMP2: .WORD   0               ;THE TEST AREA.
 4732  017570  000000                          .WORD   0
 4733
 4734
 4735  017572  032737  000002  177744  CCERR1: BIT     #2,@#MEMERR     ;SHOULD BE A MAIN MEMORY
 4736  017600  001002                          BNE     1$              ;ADDRESS AND CONTROL LINE
 4737  017602  000137  043500                  JMP     SPUR            ;PARITY ERROR.
 4738  017606  020137  177740          1$:     CMP     R1,@#LOADRS     ;ERROR ADDRESS SHOULD BE
 4739  017612  001402                          BEQ     CCERR2          ;TEST ADDRESS.
 4740  017614  000137  043500                  JMP     SPUR
 4741  017620  012637  001646          CCERR2: MOV     (SP)+,$TMP6
 4742  017624  005037  001670                  CLR     $TMP17
 4743  017630  005726                          TST     (SP)+           ;RESET THE STACK
 4744  017632  012737  000044  001672          MOV     #44,$TMP20
 4745  017640  013737  177740  001640          MOV     @#LOADRS,$TMP3
 4746  017646  013737  177742  001642          MOV     @#HIADRS,$TMP4
 4747  017654  013737  177744  001644          MOV     @#MEMERR,$TMP5
 4748  017662  010037  001646                  MOV     R0,$TMP6
 4749  017666  005037  001650                  CLR     $TMP7
 4750  017672  010037  001662                  MOV     R0,$TMP14
 4751  017676  062737  000002  001662          ADD     #2,$TMP14
 4752  017704  005037  001664                  CLR     $TMP15
 4753  017710  011037  001652                  MOV     (R0),$TMP10
 4754  017714  016037  000002  001654          MOV     2(R0),$TMP11
 4755  017722  010137  001656                  MOV     R1,$TMP12
 4756  017726  005037  001660                  CLR     $TMP13
 4757  017732  104044                  64$:    ERROR   44
 4758  017734  012737  177777  177744          MOV     #-1,@#MEMERR
 4759
 4760  017742  010002                          MOV     R0,R2
 4761  017744  020102                          CMP     R1,R2
 4762  017746  001002                          BNE     2$
 4763  017750  000137  017412                  JMP     CC4
 4764  017754  005202                  2$:     INC     R2
 4765  017756  020102                          CMP     R1,R2
 4766  017760  001002                          BNE     3$
 4767  017762  000137  017454                  JMP     CC7
 4768  017766  005202                  3$:     INC     R2
 4769  017770  020102                          CMP     R1,R2
 4770  017772  001002                          BNE     4$
 4771  017774  000137  017516                  JMP     CC10
 4772  020000  000137  020064          4$:     JMP     CCDONE
 4773
 4774
 4775  020004  011637  001656          CCERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
 4776                                                                  ;THE CORRECT BYTE
 4777  020010  010037  001636                  MOV     R0,$TMP2
 4778  020014  005037  001640                  CLR     $TMP3
 4779  020020  010037  001642                  MOV     R0,$TMP4
 4780  020024  062737  000002  001642          ADD     #2,$TMP4
 4781  020032  005037  001644                  CLR     $TMP5
 4782  020036  011037  001646                  MOV     (R0),$TMP6
```

```
4783  020042  016037  000002  001650          MOV     2(R0),$TMP7
4784  020050  010137  001652                  MOV     R1,$TMP10
4785  020054  005037  001654                  CLR     $TMP11
4786  020060  104046                          ERROR   46
4787  020062  000207                          RTS     PC
4788
4789
4790  020064  104407                  CCDONE: RSET                    ;DONE!
4791
4792
4793                                  ;;*****************************************************************
4794                                  ;*TEST 15       CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ZEROES TEST
4795                                  ;*
4796                                  ;*THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC.
4797                                  ;*HERE CPU DATOB'S WILL MOVE ZEROES INTO A BACKROUND
4798                                  ;*PATTERN OF ONES.
4799                                  ;*
4800                                  ;;*****************************************************************
4801  020066  000004                  TST15:  SCOPE
4802  020070  012737  000010  001702          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
4803          000015                  FF=$TN-1
4804                                                                  ;SET THE SKAD REGISTER
4805  020076  012737  020664  043632          MOV     #TST16,SKAD     ;IN CASE THE TEST ABORTS.
4806
4807  020104  113737  001502  001632          MOVB    $TSTNM,$TMP0
4808  020112  012737  020370  000114          MOV     #FFERR1,@#CACHVEC
4809
4810  020120  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES.
4811
4812  020126  012700  020364                  MOV     #FFTMP2,R0
4813  020132  042700  000003                  BIC     #3,R0
4814  020136  010001                          MOV     R0,R1
4815
4816  020140  012710  177777          FF1:    MOV     #-1,(R0)        ;TEST MASK 0
4817  020144  012760  177777  000002          MOV     #-1,2(R0)
4818  020152  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
4819  020154  105011                          CLRB    (R1)
4820  020156  022710  177400                  CMP     #177400,(R0)
4821  020162  001403                          BEQ     FF3
4822  020164  004737  020602          FF2:    JSR     PC,FFERR3
4823  020170  000404                          BR      FF4
4824  020172  022760  177777  000002  FF3:    CMP     #-1,2(R0)
4825  020200  001371                          BNE     FF2
4826
4827  020202  005201                  FF4:    INC     R1              ;TEST MASK 1.
4828  020204  012710  177777                  MOV     #-1,(R0)
4829  020210  012760  177777  000002          MOV     #-1,2(R0)
4830  020216  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
4831  020220  105011                          CLRB    (R1)
4832  020222  022710  000377                  CMP     #377,(R0)
4833  020226  001403                          BEQ     FF6
4834  020230  004737  020602          FF5:    JSR     PC,FFERR3
4835  020234  000404                          BR      FF7
4836  020236  022760  177777  000002  FF6:    CMP     #-1,2(R0)
4837  020244  001371                          BNE     FF5
4838
```

```
4839  020246  005201                    FF7:    INC     R1              ;TEST MASK 2.
4840  020250  012710  177777                    MOV     #-1,(R0)
4841  020254  012760  177777  000002            MOV     #-1,2(R0)
4842  020262  000240                            NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
4843  020264  105011                            CLRB    (R1)
4844  020266  022760  177400  000002            CMP     #177400,2(R0)
4845  020274  001403                            BEQ     FF9
4846  020276  004737  020602            FF8:    JSR     PC,FFERR3
4847  020302  000403                            BR      FF10
4848  020304  022710  177777            FF9:    CMP     #-1,(R0)
4849  020310  001372                            BNE     FF8
4850
4851  020312  005201                    FF10:   INC     R1              ;TEST MASK 3.
4852  020314  012710  177777                    MOV     #-1,(R0)
4853  020320  012760  177777  000002            MOV     #-1,2(R0)
4854  020326  000240                            NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
4855  020330  105011                            CLRB    (R1)
4856  020332  022760  000377  000002            CMP     #377,2(R0)
4857  020340  001403                            BEQ     FF12
4858  020342  004737  020602            FF11:   JSR     PC,FFERR3
4859  020346  000403                            BR      FF13
4860  020350  022710  177777            FF12:   CMP     #-1,(R0)
4861  020354  001372                            BNE     FF11
4862
4863  020356  000137  020662            FF13:   JMP     FFDONE
4864
4865  020362  000000                    FFTMP1: .WORD   0               ;TEST AREA.
4866  020364  000000                    FFTMP2: .WORD   0
4867  020366  000000                            .WORD   0
4868
4869
4870  020370  032737  000002  177744  FFERR1: BIT     #2,@#MEMERR     ;SHOULD BE A MAIN MEMORY
4871  020376  001002                            BNE     1$              ;ADDRESS AND CONTROL LINE
4872  020400  000137  043500                    JMP     SPUR            ;PARITY ERROR.
4873  020404  020137  177740            1$:     CMP     R1,@#LOADRS     ;. ERROR ADDRESS SHOULD BE
4874  020410  001402                            BEQ     FFERR2          ;TEST ADDRESS.
4875  020412  000137  043500                    JMP     SPUR
4876  020416  012637  001646            FFERR2: MOV     (SP)+,$TMP6
4877  020422  005037  001670                    CLR     $TMP17
4878  020426  005726                            TST     (SP)+           ;RESET THE STACK
4879  020430  012737  000050  001672            MOV     #50,$TMP20
4880  020436  013737  177740  001640            MOV     @#LOADRS,$TMP3
4881  020444  013737  177742  001642            MOV     @#HIADRS,$TMP4
4882  020452  013737  177744  001644            MOV     @#MEMERR,$TMP5
4883  020460  010037  001646                    MOV     R0,$TMP6
4884  020464  005037  001650                    CLR     $TMP7
4885  020470  010037  001662                    MOV     R0,$TMP14
4886  020474  062737  000002  001662            ADD     #2,$TMP14
4887  020502  005037  001664                    CLR     $TMP15
4888  020506  011037  001652                    MOV     (R0),$TMP10
4889  020512  016037  000002  001654            MOV     2(R0),$TMP11
4890  020520  010137  001656                    MOV     R1,$TMP12
4891  020524  005037  001660                    CLR     $TMP13
4892  020530  104050                    64$:    ERROR   50
4893  020532  012737  177777  177744            MOV     #-1,@#MEMERR
4894
```

```
4895  020540  010002                          MOV     R0,R2
4896  020542  020102                          CMP     R1,R2
4897  020544  001002                          BNE     2$
4898  020546  000137  020202                  JMP     FF4
4899  020552  005202                  2$:     INC     R2
4900  020554  020102                          CMP     R1,R2
4901  020556  001002                          BNE     3$
4902  020560  000137  020246                  JMP     FF7
4903  020564  005202                  3$:     INC     R2
4904  020566  020102                          CMP     R1,R2
4905  020570  001002                          BNE     4$
4906  020572  000137  020312                  JMP     FF10
4907  020575  000137  020662          4$:     JMP     FFDONE          ;HALT????
4908
4909
4910  020602  011637  001656          FFERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
4911                                                                  ;THE CORRECT BYTE
4912  020606  010037  001636                  MOV     R0,$TMP2
4913  020612  005037  001640                  CLR     $TMP3
4914  020616  010037  001642                  MOV     R0,$TMP4
4915  020622  062737  000002  001642          ADD     #2,$TMP4
4916  020630  005037  001644                  CLR     $TMP5
4917  020634  011037  001646                  MOV     (R0),$TMP6
4918  020640  016037  000002  001650          MOV     2(R0),$TMP7
4919  020646  010137  001652                  MOV     R1,$TMP10
4920  020652  005037  001654                  CLR     $TMP11
4921  020656  104052                          ERROR   52
4922  020660  000207                          RTS     PC
4923
4924
4925  020662  104407                  FFDONE: RSET                    ;DONE!
4926
4927                                          ;;*****************************************************************
4928                                          ;*TEST 16       CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ONES TEST
4929                                          ;*
4930                                          ;*THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC.  THIS
4931                                          ;*IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING
4932                                          ;*A WRITE.  IT DESIGNATES WHICH BYTES OF THE TWO WORDS OF
4933                                          ;*DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO
4934                                          ;*BE WRITTEN.  THIS WILL BE A TEST DOING UNIBUS DATOB REFERENCES TO
4935                                          ;*THE CACHE.  THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN
4936                                          ;*OF ZEROES.
4937                                          ;*
4938                                          ;;*****************************************************************
4939  020664  000004                  TST16:  SCOPE
4940  020666  012737  000010  001702          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
4941          000016                  EE=$^N-1
4942                                                                  ;SET THE SKAD REGISTER
4943  020674  012737  021552  043632          MOV     #TST17,SKAD     ;IN CASE THE TEST ABORTS.
4944
4945  020702  113737  001502  001632          MOVB    $TSTNM,$TMP0
4946  020710  104411                          MMSKIP
4947  020712  012737  021256  000114          MOV     #EEERR1,@#CACHVEC
4948
4949  020720  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
4950                                                                  ;TO RELOCATE EVERYTHING
```

```
4951  020724  012702  172300              MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
4952  020730  012703  000007              MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
4953  020734  005004                      CLR     R4              ;BY PASSIVELY IS MEANT
4954  020736  012705  170200              MOV     #MAPL00,R5      ;THAT ADDRESS ARE
4955                                                              ;RELOCATED TO THEMSELVES.
4956  020742  012722  077406      64$:    MOV     #77406,(R2)+
4957  020746  010401                      MOV     R4,R1
4958  020750  072127  000006              ASH     #6,R1
4959  020754  010125                      MOV     R1,(R5)+
4960  020756  005025                      CLR     (R5)+
4961  020760  010410                      MOV     R4,(R0)
4962  020762  062720  170000              ADD     #170000,(R0)+
4963  020766  062704  000200              ADD     #200,R4
4964  020772  077315                      SOB     R3,64$
4965  020774  012710  177600              MOV     #177600,(R0)
4966  021000  012712  077406              MOV     #77406,(R2)
4967
4968  021004  012737  000060  172516      MOV     #60,@#MMR3      ;TURN ON MEMORY MANAGEMENT
4969  021012  012737  000001  177572      MOV     #1,@#MMR0       ;AND THE MAPPING BOX RELOCATION.
4970
4971  021020  012737  000014  177746      MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
4972
4973  021026  012700  021252              MOV     #EETMP2,R0      ;LOCATE THE TEST SPACE.
4974  021032  042700  000003              BIC     #3,R0
4975  021036  010001                      MOV     R0,R1
4976
4977  021040  005010              EE1:    CLR     (R0)            ;TEST MASK 0
4978  021042  005060  000002              CLR     2(R0)
4979  021046  000240                      NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
4980  021050  112711  000377              MOVB    #377,(R1)
4981  021054  022710  000377              CMP     #377,(R0)
4982  021060  001403                      BEQ     EE3
4983  021062  004737  021470      EE2:    JSR     PC,EEERR3
4984  021066  000403                      BR      EE4
4985  021070  005760  000002      EE3:    TST     2(R0)
4986  021074  001372                      BNE     EE2
4987
4988  021076  062701  000001      EE4:    ADD     #1,R1
4989  021102  005010                      CLR     (R0)
4990  021104  005060  000002              CLR     2(R0)
4991  021110  000240                      NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
4992  021112  112711  000377              MOVB    #377,(R1)
4993  021116  022710  177400              CMP     #177400,(R0)
4994  021122  001403                      BEQ     EE6
4995  021124  004737  021470      EE5:    JSR     PC,EEERR3
4996  021130  000403                      BR      EE7
4997  021132  005760  000002      EE6:    TST     2(R0)
4998  021136  001372                      BNE     EE5
4999
5000  021140  062701  000001      EE7:    ADD     #1,R1
5001  021144  005010                      CLR     (R0)
5002  021146  005060  000002              CLR     2(R0)
5003  021152  000240                      NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
5004  021154  112711  000377              MOVB    #377,(R1)
5005  021160  022760  000377  000002      CMP     #377,2(R0)
5006  021166  001403                      BEQ     EE9
```

```
5007  021170  004737  021470          EE8:    JSR     PC,EEERR3
5008  021174  000402                  EE10    BR      EE10
5009  021176  005710                  EE9:    TST     (R0)
5010  021200  001373                  EE8     BNE     EE8
5011
5012  021202  062701  000001          EE10:   ADD     #1,R1
5013  021206  005010                  EE10    CLR     (R0)
5014  021210  005060  000002          EE10    CLR     2(R0)
5015  021214  000240                  EE10    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
5016  021216  112711  000377          EE10    MOVB    #377,(R1)
5017  021222  022760  177400  000002  EE10    CMP     #177400,2(R0)
5018  021230  001403                  EE10    BEQ     EE12
5019  021232  004737  021470          EE11:   JSR     PC,EEERR3
5020  021236  000402                  EE11    BR      EE13
5021  021240  005710                  EE12:   TST     (R0)
5022  021242  001373                  EE12    BNE     EE11
5023
5024  021244  000137  021550          EE13:   JMP     EEDONE
5025
5026  021250  000000                  EETMP1: .WORD   0
5027  021252  000000                  EETMP2: .WORD   0
5028  021254  000000                  EE10    .WORD   0
5029
5030
5031  021256  032737  000002  177744  EEERR1: BIT     #2,@#MEMERR     ;SHOULD BE A MAIN MEMORY
5032  021264  001002                  EEERR1  BNE     1$              ;ADDRESS AND CONTROL LINE
5033  021266  000137  043500          EEERR1  JMP     SPUR            ;PARITY ERROR.
5034  021272  020137  177740          1$:     CMP     R1,@#LOADRS     ;ERROR ADDRESS SHOULD BE
5035  021276  001402                  EEERR1  BEQ     EEERR2          ;TEST ADDRESS.
5036  021300  000137  043500          EEERR1  JMP     SPUR
5037  021304  012637  001646          EEERR2: MOV     (SP)+,$TMP6
5038  021310  005037  001670          EEERR2  CLR     $TMP17
5039  021314  005726                  EEERR2  TST     (SP)+           ;RESET THE STACK
5040  021316  012737  000045  001672  EEERR2  MOV     #45,$TMP20
5041  021324  013737  177740  001640  EEERR2  MOV     @#LOADRS,$TMP3
5042  021332  013737  177742  001642  EEERR2  MOV     @#HIADRS,$TMP4
5043  021340  013737  177744  001644  EEERR2  MOV     @#MEMERR,$TMP5
5044  021346  010037  001646          EEERR2  MOV     R0,$TMP6
5045  021352  005037  001650          EEERR2  CLR     $TMP7
5046  021356  010037  001662          EEERR2  MOV     R0,$TMP14
5047  021362  062737  000002  001662  EEERR2  ADD     #2,$TMP14
5048  021370  005037  001664          EEERR2  CLR     $TMP15
5049  021374  011037  001652          EEERR2  MOV     (R0),$TMP10
5050  021400  016037  000002  001654  EEERR2  MOV     2(R0),$TMP11
5051  021406  010137  001656          EEERR2  MOV     R1,$TMP12
5052  021412  005037  001660          EEERR2  CLR     $TMP13
5053  021416  104045                  64$:    ERROR   45
5054  021420  012737  177777  177744  EEERR2  MOV     #-1,@#MEMERR
5055
5056  021426  010002                  EEERR2  MOV     R0,R2
5057  021430  020102                  EEERR2  CMP     R1,R2
5058  021432  001002                  EEERR2  BNE     2$
5059  021434  000137  021076          EEERR2  JMP     EE4
5060  021440  005202                  2$:     INC     R2
5061  021442  020102                  EEERR2  CMP     R1,R2
5062  021444  001002                  EEERR2  BNE     3$
```

```
5063  021446  000137  021140              JMP     EE7
5064  021452  005202              3$:     INC     R2
5065  021454  020102                      CMP     R1,R2
5066  021456  001002                      BNE     4$
5067  021460  000137  021202              JMP     EE10
5068  021464  000137  021550      4$:     JMP     EEDONE
5069
5070
5071  021470  011637  001656      EEERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
5072                                                              ;THE CORRECT BYTE
5073  021474  010037  001636              MOV     R0,$TMP2
5074  021500  005037  001640              CLR     $TMP3
5075  021504  010037  001642              MOV     R0,$TMP4
5076  021510  062737  000002  001642      ADD     #2,$TMP4
5077  021516  005037  001644              CLR     $TMP5
5078  021522  011037  001646              MOV     (R0),$TMP6
5079  021526  016037  000002  001650      MOV     2(R0),$TMP7
5080  021534  010137  001652              MOV     R1,$TMP10
5081  021540  005037  001654              CLR     $TMP11
5082  021544  104047                      ERROR   47
5083  021546  000207                      RTS     PC
5084
5085
5086  021550  104407      EEDONE: RSET                    ;DONE!
5087
5088                      ;;*****************************************************************
5089                      ;*TEST 17       CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ZEROES TEST
5090                      ;*
5091                      ;*THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC.
5092                      ;*HERE UNIBUS DATOB'S WILL MOVE ZEROES INTO A BACKROUND
5093                      ;*PATTERN OF ONES.
5094                      ;*
5095                      ;*****************************************************************
5096  021552  000004      TST17:  SCOPE
5097  021554  012737  000010  001702      MOV     #10,$TIMES      ;;DO 10 ITERATIONS
5098          000017      HH=$TN-1
5099                                                              ;SET THE SKAD REGISTER
5100  021562  012737  022452  043632      MOV     #TST20,SKAD     ;IN CASE THE TEST ABORTS.
5101
5102  021570  113737  001502  001632      MOVB    $TSTNM,$TMP0
5103
5104  021576  104411                      MMSKIP
5105
5106  021600  012737  022156  000114      MOV     #HHERR1,@#CACHVEC
5107
5108
5109  021606  012700  172340              MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
5110                                                              ;TO RELOCATE EVERYTHING
5111  021612  012702  172300              MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
5112  021616  012703  000007              MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
5113  021622  005004                      CLR     R4              ;BY PASSIVELY IS MEANT
5114  021624  012705  170200              MOV     #MAPLOD,R5      ;THAT ADDRESS ARE
5115                                                              ;RELOCATED TO THEMSELVES.
5116  021630  012722  077406      64$:    MOV     #77406,(R2)+
5117  021634  010401                      MOV     R4,R1
5118  021636  072127  000006              ASH     #6,R1
```

```
5119  021642  010125                          MOV     R1,(R5)+
5120  021644  005025                          CLR     (R5)+
5121  021646  010410                          MOV     R4,(R0)
5122  021650  062720  170000                  ADD     #170000,(R0)+
5123  021654  062704  000200                  ADD     #200,R4
5124  021660  077315                          SOB     R3,64$
5125  021662  012710  177600                  MOV     #177600,(R0)
5126  021666  012712  077406                  MOV     #77406,(R2)
5127
5128  021672  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON MEMORY MANAGEMENT
5129  021700  012737  000001  177572          MOV     #1,@#MMR0       ;AND MAPPING BOX RELOCATION.
5130
5131  021706  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES.
5132
5133  021714  012700  022152                  MOV     #HHTMP2,R0      ;LOCATE THE TEST SPACE.
5134  021720  042700  000003                  BIC     #3,R0
5135  021724  010001                          MOV     R0,R1
5136
5137  021726  012710  177777          HH1:    MOV     #-1,(R0)
5138  021732  012760  177777  000002          MOV     #-1,2(R0)
5139  021740  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
5140  021742  105011                          CLRB    (R1)
5141  021744  022710  177400                  CMP     #177400,(R0)
5142  021750  001403                          BEQ     HH3
5143  021752  004737  022370          HH2:    JSR     PC,HHERR3
5144  021756  000404                          BR      HH4
5145  021760  022760  177777  000002  HH3:    CMP     #-1,2(R0)
5146  021766  001371                          BNE     HH2
5147
5148  021770  005201                  HH4:    INC     R1
5149  021772  012710  177777                  MOV     #-1,(R0)
5150  021776  012760  177777  000002          MOV     #-1,2(R0)
5151  022004  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
5152  022006  105011                          CLRB    (R1)
5153  022010  022710  000377                  CMP     #377,(R0)
5154  022014  001403                          BEQ     HH6
5155  022016  004737  022370          HH5:    JSR     PC,HHERR3
5156  022022  000404                          BR      HH7
5157  022024  022760  177777  000002  HH6:    CMP     #-1,2(R0)
5158  022032  001371                          BNE     HH5
5159
5160  022034  005201                  HH7:    INC     R1
5161  022036  012710  177777                  MOV     #-1,(R0)
5162  022042  012760  177777  000002          MOV     #-1,2(R0)
5163  022050  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
5164  022052  105011                          CLRB    (R1)
5165  022054  122760  177400  000002          CMPB    #177400,2(R0)
5166  022062  001403                          BEQ     HH9
5167  022064  004737  022370          HH8:    JSR     PC,HHERR3
5168  022070  000403                          BR      HH10
5169  022072  022710  177777          HH9:    CMP     #-1,(R0)
5170  022076  001372                          BNE     HH8
5171
5172  022100  005201                  HH10:   INC     R1
5173  022102  012710  177777                  MOV     #-1,(R0)
5174  022106  012760  177777  000002          MOV     #-1,2(R0)
```

```
5175  022114  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
5176  022116  105011                          CLRB    (R1)
5177  022120  022760  000377  000002          CMP     #377,2(R0)
5178  022126  001403                          BEQ     HH12
5179  022130  004737  022370          HH11:   JSR     PC,HHERR3
5180  022134  000403                          BR      HH13
5181  022136  022710  177777          HH12:   CMP     #-1,(R0)
5182  022142  001372                          BNE     HH11
5183
5184  022144  000137  022450          HH13:   JMP     HHDONE
5185
5186  022150  000000                  HHTMP1: .WORD   0
5187  022152  000000                  HHTMP2: .WORD   0               ;THE TEST AREA
5188  022154  000000                          .WORD   0
5189
5190
5191  022156  032737  000002  177744  HHERR1: BIT     #2,@#MEMERR     ;SHOULD BE A MAIN MEMORY
5192  022164  001002                          BNE     1$              ;ADDRESS AND CONTROL LINE
5193  022166  000137  043500                  JMP     SPUR            ;PARITY ERROR.
5194  022172  020137  177740          1$:     CMP     R1,@#LOADRS     ;ERROR ADDRESS SHOULD BE
5195  022176  001402                          BEQ     HHERR2          ;TEST ADDRESS.
5196  022200  000137  043500                  JMP     SPUR
5197  022204  012637  001646          HHERR2: MOV     (SP)+,$TMP6
5198  022210  005037  001670                  CLR     $TMP17
5199  022214  005726                          TST     (SP)+           ;RESET THE STACK
5200  022216  012737  000051  001672          MOV     #51,$TMP20
5201  022224  013737  177740  001640          MOV     @#LOADRS,$TMP3
5202  022232  013737  177742  001642          MOV     @#HIADRS,$TMP4
5203  022240  013737  177744  001644          MOV     @#MEMERR,$TMP5
5204  022246  010037  001646                  MOV     R0,$TMP6
5205  022252  005037  001650                  CLR     $TMP7
5206  022256  010037  001662                  MOV     R0,$TMP14
5207  022262  062737  000002  001662          ADD     #2,$TMP14
5208  022270  005037  001664                  CLR     $TMP15
5209  022274  011037  001652                  MOV     (R0),$TMP10
5210  022300  016037  000002  001654          MOV     2(R0),$TMP11
5211  022306  010137  001656                  MOV     R1,$TMP12
5212  022312  005037  001660                  CLR     $TMP13
5213  022316  104051                  64$:    ERROR   51
5214  022320  012737  177777  177744          MOV     #-1,@#MEMERR
5215
5216  022326  010002                          MOV     R0,R2
5217  022330  020102                          CMP     R1,R2
5218  022332  001002                          BNE     2$
5219  022334  000137  021770                  JMP     HH4
5220  022340  005202                  2$:     INC     R2
5221  022342  020102                          CMP     R1,R2
5222  022344  001002                          BNE     3$
5223  022346  000137  022034                  JMP     HH7
5224  022352  005202                  3$:     INC     R2
5225  022354  020102                          CMP     R1,R2
5226  022356  001002                          BNE     4$
5227  022360  000137  022100                  JMP     HH10
5228  022364  000137  022450          4$:     JMP     HHDONE
5229
5230
```

```
5231  022370  011637  001656          HHERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
5232                                                                  ;THE CORRECT BYTE
5233  022374  010037  001636                  MOV     R0,$TMP2
5234  022400  005037  001640                  CLR     $TMP3
5235  022404  010037  001642                  MOV     R0,$TMP4
5236  022410  062737  000002  001642          ADD     #2,$TMP4
5237  022416  005037  001644                  CLR     $TMP5
5238  022422  011037  001646                  MOV     (R0),$TMP6
5239  022426  016037  000002  001650          MOV     2(R0),$TMP7
5240  022434  010137  001652                  MOV     R1,$TMP10
5241  022440  005037  001654                  CLR     $TMP11
5242  022444  104053                          ERROR   53
5243  022446  000207                          RTS     PC
5244
5245
5246  022450  104407          HHDONE: RSET                    ;DONE!
5247
5248
5249                          ;;*****************************************************************
5250                          ;*TEST 20       CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST
5251                          ;*
5252                          ;*THIS TEST IS EXECUTED OPTIONALLY, ON THE CONDITION THAT
5253                          ;*BIT 12 OF THE SWITCH REGISTER IS ON WHEN PROGRAM CONTROL
5254                          ;*REACHES THIS POINT.  IF THIS SWITCH IS OFF THEN CONTROL
5255                          ;*IS PASSED TO THE NEXT TEST.  THIS IS DONE BECAUSE THIS
5256                          ;*TEST REQUIRES OPERATOR INTERVENTION.  THE USER IS ASKED TO
5257                          ;*GO THROUGH A POWER DOWN-POWER UP SEQUENCE.  THEN
5258                          ;*A SIMPLE SCAN IS MADE OF MEMORY WHICH CAUSES ALL
5259                          ;*DATA AND ADDRESS MEMORY LOCATIONS IN THE CACHE TO BE
5260                          ;*PARITY CHECKED.  IF THE POWER UP-CACHE INVLIDATER LOGIC
5261                          ;*WORKED NO PARITY ERRORS CAN OCCUR.  BUT IF THIS INVALIDATER
5262                          ;*FAILED THERE IS AN EXTREMELY HIGH PROBABILITY FOR THE
5263                          ;*OCCURENCE OF A CACHE DATA OR CACHE ADDRESS PARITY ERROR.
5264                          ;*IN FACT IF THE INVALIDATER CIRCUIT IS COMPLETELY INOPERATIVE
5265                          ;*IT WILL BE VIRTUALLY IMPOSSIBLE TO RESTART THE PROGRAM.
5266                          ;*WHEREAS MINOR OR NO FAILURES CAN AND WILL BE REPORTED.
5267                          ;*IF NO PARITY ERRORS ARE ENCOUNTERED THE USER WILL
5268                          ;*BE NOTIFIED SO THAT HE CAN KNOW IF A FATAL FAILURE
5269                          ;*HAS OCCURRED.
5270                          ;*
5271                          ;;*****************************************************************
5272  022452  000004          TST20:  SCOPE
5273          000020          DD=$TN-1
5274                                                          ;SET THE SKAD REGISTER
5275  022454  012737  022706  043632          MOV     #TST21,SKAD     ;IN CASE THE TEST ABORTS.
5276
5277  022462  113737  001502  001632          MOVB    $TSTNM,$TMP0
5278  022470  012737  043500  000114          MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
5279
5280  022476  032777  010000  157034          BIT     #SW12,@SWR      ;SEE IF THE USER HAS CHOSEN
5281  022504  001002                          BNE     DD1             ;TO RUN THIS TEST, SW12=1.
5282  022506  000177  021120                  JMP     @SKAD           ;NO, SO GO TO NEXT TEST.
5283
5284  022512  012737  022644  000114  DD1:    MOV     #DDPER,@#CACHVEC        ;YES, SO SET UP THE PARITY
5285                                                          ;ERROR VECTOR.
5286  022520  013737  000024  022674          MOV     @#24,DDTMP      ;SAVE THE OLD CONTENTS
```

```
5287  022526  012737  022546  000024          MOV     #DDPD,@#24      ;OF THE PWER FAIL TRAP
5288  022534  005037  022676                  CLR     DDCNTR          ;VECTOR AND RESET THIS
5289                                                                  ;VECTOR. CLEAR AN ERROR COUNT.
5290  022540  104400                          TYPE                    ;TELL THE USER TO POWER
5291  022542  053120                          .WORD   PDMSG1          ;DOWN.
5292  022544  000777                          BR      .               ;WAIT, SHOULD THIS
5293                                                                  ;WAIT TIME OUT????
5294  022546  000240                  DDPD:   NOP                     ;FOR SCOPE SYNC!
5295  022550  012737  022560  000024          MOV     #DDPV,@#24      ;POWER DOWN ROUTINE
5296  022556  000777                          BR      .               ;JUST SET UP FOR POWER UP.
5297  022560  012706  001500          DDPV:   MOV     #STACK,SP       ;RESET THE STACK POINTER
5298  022564  013737  022674  000024          MOV     DDTMP,@#24      ;RESET POWER FAIL VECTOR.
5299  022572  005000                          CLR     R0              ;SET UP FOR SCAN.
5300  022574  012701  001000                  MOV     #1000,R1
5301  022600  005720                  1$:     TST     (R0)+
5302  022602  077102                          SOB     R1,1$
5303  022604  013737  022674  000024  DDPU1:  MOV     DDTMP,@#24      ;RESET THE POWER FAIL VECTOR.
5304  022612  005737  022676                  TST     DDCNTR          ;WERE THERE ANY ERRORS?
5305  022616  001004                          BNE     DDPU2
5306  022620  104400                          TYPE                    ;NO
5307  022622  053276                          .WORD   PDMSG2
5308  022624  000137  022700                  JMP     DDDONE
5309
5310  022630                          DDPU2:                          ;REPORT ERROR SUMMARY
5311  022630  013737  022676  001636          MOV     DDCNTR,$TMP2
5312  022636  104054                  1$:     ERROR   54
5313  022640  000137  022700                  JMP     DDDONE
5314
5315  022644  032737  000360  177744  DDPER:  BIT     #360,@#MEMERR   ;THE ERROR SHOULD BE
5316  022652  001406                          BEQ     DDPER1          ;A CACHE ADDRESS OR CACHE
5317  022654  012737  177777  177744          MOV     #-1,MEMERR      ;DATA PARITY ERROR
5318  022662  005237  022676                  INC     DDCNTR
5319  022666  000002                          RTI
5320
5321  022670  000137  043500          DDPER1: JMP     SPUR
5322
5323  022674  000000                  DDTMP:  .WORD   0               ;STORAGE FOR POWER FAIL
5324                                                                  ;VECTORS OLD PC
5325  022676  000000                  DDCNTR: .WORD   0               ;ERROR COUNT.
5326
5327  022700  104407                  DDDONE: RSET
5328  022702  012706  001500                  MOV     #STACK,SP
5329
5330                                  ;;*****************************************************************
5331                                  ;*TEST 21        CACHE DATA MULTIPLEXER, CDMX, TEST
5332                                  ;*
5333                                  ;*THIS TEST PUTS DIFFERENT PATTERNS OF DATA AT THE INPUTS
5334                                  ;*OF THE CDMX AND TESTS FOR PROPER SELECTION AND GOOD DATA.
5335                                  ;*
5336                                  ;;*****************************************************************
5337  022706  000004                  TST21:  SCOPE
5338  022710  012737  000010  001702          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
5339                                                                  ;SET THE SKAD REGISTER
5340  022716  012737  024014  043632          MOV     #TST22,SKAD     ;IN CASE THE TEST ABORTS.
5341
5342  022724  012737  043500  000114          MOV     #SPUR,@#CACHVEC ;PREPARE FOR UNEXPECTED ERRORS.
```

```
5343  022732  113737  001502  001632          MOVB    $TSTNM,$TMP0
5344  022740  012705  000006                  MOV     #6,R5           ;INITIALIZE
5345  022744  012737  000004  023766          MOV     #4,JJCNT
5346  022752  012700  024004                  MOV     #JJTMP2,R0
5347  022756  042700  176002                  BIC     #176002,R0
5348  022762  012701  140000                  MOV     #TESTR1,R1
5349  022766  060001                          ADD     R0,R1
5350  022770  012702  142000                  MOV     #TESTR2,R2
5351  022774  060002                          ADD     R0,R2
5352  022776  012703  144000                  MOV     #TESTR3,R3
5353  023002  060003                          ADD     R0,R3
5354  023004  012704  023772                  MOV     #JJPAT2,R4
5355
5356  023010  012737  125252  023770          MOV     #125252,JJPAT1  ;JJPAT1 CONTAINS THE DATA
5357                                                                  ;WHICH WILL ENTER THE
5358                                                                  ;MAIN MEMORY EVEN INPUTS
5359                                                                  ;TO THE CDMX. INITIALLY
5360                                                                  ;THIS WILL BE 125252
5361  023016  012737  052525  023772          MOV     #52525,JJPAT2   ;DATA FOR MAIN MEMORY ODD
5362                                                                  ;WORD INPUT TO CDMX
5363  023024  005037  023774                  CLR     JJPAT3          ;GROUP 0 DATA INPUTS TO CDMX.
5364  023030  012737  177777  023776          MOV     #-1,JJPAT4      ;GROUP 1 DATA INPUTS TO CDMX.
5365  023036  012737  023036  001510  JJ1:    MOV     #JJ1,$LPERR
5366  023044  013713  023770                  MOV     JJPAT1,(R3)     ;WRITE THE MAIN MEMORY
5367  023050  013763  023772  000002          MOV     JJPAT2,2(R3)    ;EVEN AND ODD WORD PATTERNS
5368
5369  023056  012737  000034  177746          MOV     #SOMOM1,@#CONTRL        ;WRITE THE GROUP ZERO
5370  023064  013711  023774                  MOV     JJPAT3,(R1)     ;PATTERN
5371  023070  013761  023774  177776          MOV     JJPAT3,-2(R1)
5372  023076  013761  023774  000002          MOV     JJPAT3,2(R1)
5373  023104  005711                          TST     (R1)
5374  023106  012737  000054  177746          MOV     #S1MOM1,@#CONTRL        ;WRITE THE GROUP ONE PATTERN
5375  023114  013712  023776                  MOV     JJPAT4,(R2)
5376  023120  013762  023776  177776          MOV     JJPAT4,-2(R2)
5377  023126  013762  023776  000002          MOV     JJPAT4,2(R2)
5378  023134  005712                          TST     (R2)
5379
5380  023136  005037  177746                  CLR     @#CONTRL
5381  023142  000240                          NOP
5382  023144                          JJ2:
5383  023144  000240                          NOP
5384  023146  016100  000000                  MOV     0(R1),R0
5385  023152  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
5386  023160  001011                          BNE     JJ3
5387  023162  012737  000000  001634          MOV     #0,$TMP1
5388  023170  010137  001636                  MOV     R1,$TMP2
5389  023174  062737  000000  001636          ADD     #0,$TMP2
5390  023202  104001                  66$:    ERROR   1
5391  023204  020037  023774          JJ3:    CMP     R0,JJPAT3
5392  023210  001406                          BEQ     65$
5393  023212  012737  023224  001634          MOV     #64$,$TMP1
5394  023220  010037  001636                  MOV     R0,$TMP2
5395  023224  104005                  64$:    ERROR   5
5396  023226                          65$:
5397  023226  012737  023234  001510          MOV     #JJ4,$LPERR
5398  023234                          JJ4:
```

```
5399  023234  000240                          NOP
5400  023236  016100  000002                  MOV     2(R1),R0
5401  023242  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
5402  023250  001011                          BNE     JJ5
5403  023252  012737  000000  001634          MOV     #0,$TMP1
5404  023260  010137  001636                  MOV     R1,$TMP2
5405  023264  062737  000002  001636          ADD     #2,$TMP2
5406  023272  104001                  66$:    ERROR   1
5407  023274  020037  023774          JJ5:    CMP     R0,JJPAT3
5408  023300  001406                          BEQ     65$
5409  023302  012737  023314  001634          MOV     #64$,$TMP1
5410  023310  010037  001636                  MOV     R0,$TMP2
5411  023314  104005                  64$:    ERROR   5
5412  023316                          65$:
5413  023316  012737  023324  001510          MOV     #JJ6,$LPERR
5414  023324                          JJ6:
5415  023324  000240                          NOP
5416  023326  016200  000000                  MOV     0(R2),R0
5417  023332  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
5418  023340  001011                          BNE     JJ7
5419  023342  012737  000001  001634          MOV     #1,$TMP1
5420  023350  010237  001636                  MOV     R2,$TMP2
5421  023354  062737  000000  001636          ADD     #0,$TMP2
5422  023362  104001                  66$:    ERROR   1
5423  023364  020037  023776          JJ7:    CMP     R0,JJPAT4
5424  023370  001406                          BEQ     65$
5425  023372  012737  023404  001634          MOV     #64$,$TMP1
5426  023400  010037  001636                  MOV     R0,$TMP2
5427  023404  104006                  64$:    ERROR   6
5428  023406                          65$:
5429  023406  012737  023414  001510          MOV     #JJ8,$LPERR
5430  023414                          JJ8:
5431  023414  000240                          NOP
5432  023416  016200  000002                  MOV     2(R2),R0
5433  023422  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
5434  023430  001011                          BNE     JJ9
5435  023432  012737  000001  001634          MOV     #1,$TMP1
5436  023440  010237  001636                  MOV     R2,$TMP2
5437  023444  062737  000002  001636          ADD     #2,$TMP2
5438  023452  104001                  66$:    ERROR   1
5439  023454  020037  023776          JJ9:    CMP     R0,JJPAT4
5440  023460  001406                          BEQ     65$
5441  023462  012737  023474  001634          MOV     #64$,$TMP1
5442  023470  010037  001636                  MOV     R0,$TMP2
5443  023474  104006                  64$:    ERROR   6
5444  023476                          65$:
5445  023476  012737  023504  001510          MOV     #JJ10,$LPERR
5446  023504  000240                  JJ10:   NOP
5447  023506  012737  000014  177746          MOV     #M1M0,@#CONTRL  ;CHECK MAIN MEMORY DATA
5448  023514  011300                          MOV     (R3),R0         ;EVEN WORD
5449  023516  020037  023770                  CMP     R0,JJPAT1
5450  023522  001403                          BEQ     1$
5451  023524  010037  001636                  MOV     R0,$TMP2
5452  023530  104007                          ERROR   7
5453  023532  012737  023540  001510  1$:     MOV     #JJ11,$LPERR
5454  023540  016300  000002          JJ11:   MOV     2(R3),R0        ;CHECK MAIN MEMORY EVEN
```

```
5455  023544  020037  023772              CMP     R0,JJPAT2       ;WORD
5456  023550  001403                      BEQ     JJ12
5457  023552  010037  001636              MOV     R0,$TMP2
5458  023556  104010              1$:     ERROR   10
5459
5460  023560  005037  177746      JJ12:   CLR     @#CONTRL
5461  023564  020427  023776              CMP     R4,#JJPAT4      ;NOW GET EVERY PERMUTATION
5462  023570  001011                      BNE     JJ13            ;OF THE FOUR TEST PATTERNS:
5463                                                              ;125252,052525,177777 AND
5464  023572  011437  024000              MOV     (R4),JJPAT5     ;000000 INTO JJPAT1, JJPAT2,
5465  023576  013714  023772              MOV     JJPAT2,(R4)     ;JJPAT3 AND JJPAT4 AND
5466  023602  012704  023772              MOV     #JJPAT2,R4      ;REPEAT THE TEST.
5467  023606  013714  024000              MOV     JJPAT5,(R4)
5468  023612  000406                      BR      JJ14
5469
5470  023614  012437  024000      JJ13:   MOV     (R4)+,JJPAT5
5471  023620  011464  177776              MOV     (R4),-2(R4)
5472  023624  013714  024000              MOV     JJPAT5,(R4)
5473
5474  023630  005305              JJ14:   DEC     R5
5475  023632  001402                      BEQ     1$
5476  023634  000137  023036              JMP     JJ1
5477  023640  012705  000006      1$:     MOV     #6,R5
5478  023644  013737  023770  024000      MOV     JJPAT1,JJPAT5
5479  023652  005337  023766              DEC     JJCNT
5480
5481  023656  023727  023766  000003      CMP     JJCNT,#3
5482  023664  001010                      BNE     JJ15
5483  023666  013737  023772  023770      MOV     JJPAT2,JJPAT1
5484  023674  013737  024000  023772      MOV     JJPAT5,JJPAT2
5485  023702  000137  023036              JMP     JJ1
5486
5487  023706  023727  023766  000002  JJ15:   CMP     JJCNT,#2
5488  023714  001010                      BNE     JJ16
5489  023716  013737  023774  023770      MOV     JJPAT3,JJPAT1
5490  023724  013737  024000  023774      MOV     JJPAT5,JJPAT3
5491  023732  000137  023036              JMP     JJ1
5492
5493  023736  023727  023766  000001  JJ16:   CMP     JJCNT,#1
5494  023744  001023                      BNE     JJ17            ;DONE?
5495  023746  013737  023776  023770      MOV     JJPAT4,JJPAT1
5496  023754  013737  024000  023776      MOV     JJPAT5,JJPAT4
5497  023762  000137  023036              JMP     JJ1
5498
5499  023766  000000              JJCNT:  .WORD   0               ;COUNTER USED TO GENERATE
5500                                                              ;PERMUTATIONS.
5501  023770  000000              JJPAT1: .WORD   0               ;MAIN MEMORY EVEN WORD DATA PATTERN
5502  023772  000000              JJPAT2: .WORD   0               ;MAIN MEMORY ODD WORD DATA PATTERN
5503  023774  000000              JJPAT3: .WORD   0               ;GROUP 0 DATA PATTERN
5504  023776  000000              JJPAT4: .WORD   0               ;GROUP 1 DATA PATTERN
5505  024000  000000              JJPAT5: .WORD   0               ;TEMPORARY STORAGE
5506
5507  024002  000000              JJTMP1: .WORD   0               ;TEST AREA, SO CODE WON'T
5508  024004  000000  000000  000000  JJTMP2: .WORD   0,0,0,0     ;OVER LAP THE HITS OF
5509  024012  000000
5510                                                              ;THE TEST WORDS.
```

```
        024014                                  JJ17:                                   ;DONE!
                                                ;;*******************************************************************
                                                ;*TEST 22        CACHE DATA MEMORY ADDRESS DRIVERS TEST
                                                ;*
                                                ;*THIS TEST PERFORMS A DUAL ADDRESS TEST ON THE
                                                ;*CACHE DATA MEMORIES OF BOTH GROUPS.
                                                ;*
                                                ;;*******************************************************************
        024014  000004                          TST22:  SCOPE
        024016  012737  000010  001702                  MOV     #10,$TIMES              ;;DO 10 ITERATIONS
                                                                                        ;SET THE SKAD REGISTER
        024024  012737  024524  043632                  MOV     #TST23,SKAD             ;IN CASE THE TEST ABORTS.
        024032  012737  043500  000114                  MOV     #SPUR,@#CACHVEC
        024040  113737  001502  001632                  MOVB    $TSTNM,$TMP0
        024046  012737  000001  024516  GG1:            MOV     #1,GGFLG1               ;INITIALIZE FOR A TEST
        024054  012737  000054  024520                  MOV     #S1M0M1,GGGS            ;ON GROUP 1 FIRST
        024062  012737  000034  024522                  MOV     #S0M0M1,GGGM            ;S0M1 AND S1M0 ARE PATTERNS
                                                                                        ;DESTINED FOR THE CACHE
                                                                                        ;CONTROL REGISTER
        024070  012700  024070          GG2:            MOV     #GG2,R0                 ;MAKE THIS CODE, LOCATIONS
        024074  012701  001000                          MOV     #1000,R1                ;GG2 THROUGH GG2+2000(OCT),
        024100  013737  024520  177746  GG3:            MOV     GGGS,@#CONTRL           ;HITS IN THE GROUP NOT
        024106  005760  002000                          TST     2000(R0)                ;BEING TESTED AND MISSES
        024112  013737  024522  177746                  MOV     GGGM,@#CONTRL           ;IN THE GROUP BEING TESTED.
        024120  005720                                  TST     (R0)+
        024122  077112                                  SOB     R1,GG3
        024124  013700  024520                          MOV     GGGS,R0                 ;MAKE THE TEST AREA
        024130  042700  177717                          BIC     #177717,R0              ;HITS IN THE GROUP
        024134  010037  177746                          MOV     R0,@#CONTRL             ;BEING TESTED
        024140  012701  140000                          MOV     #TESTR1,R1
        024144  012700  001000                          MOV     #1000,R0
        024150  012737  024156  001510                  MOV     #GG4,$LPERR
        024156  000240                          GG4:    NOP
        024160  005011                                  CLR     (R1)
        024162  005711                                  TST     (R1)
        024164  005711                                  TST     (R1)
        024166  032737  000010  177752                  BIT     #10,@#HITMISS
        024174  001006                                  BNE     2$
        024176  013737  024516  001634                  MOV     GGFLG1,$TMP1
        024204  010137  001636                          MOV     R1,$TMP2
        024210  104001                          1$:     ERROR   1
        024212  005721                          2$:     TST     (R1)+
        024214  077020                                  SOB     R0,GG4
        024216  013700  024522                          MOV     GGGM,R0                 ;FROM HERE ON SELECT
        024222  042700  177717                          BIC     #177717,R0              ;THE GROUP NOT BEING
        024226  010037  177746                          MOV     R0,@#CONTRL             ;TESTED
        024232  012701  140000                          MOV     #TESTR1,R1
        024236  012700  001000                          MOV     #1000,R0
        024242  012737  024250  001510                  MOV     #GG5,$LPERR
        024250  000240                          GG5:    NOP                             ;
        024252  010111                                  MOV     R1,(R1)                 ;WRITE #ADDRESS INTO @#ADDRESS.
```

```
5567  024254  005721                          TST     (R1)+
5568  024256  077004                          SOB     R0,GG5
5569
5570  024260  012701  140000                  MOV     #TESTR1,R1
5571  024264  012700  001000                  MOV     #1000,R0
5572  024270  012737  024276  001510          MOV     #GG6,$LPERR
5573  024276  000240                  GG6:    NOP
5574  024300  011102                          MOV     (R1),R2         ;READ BACK THE ADDRESS
5575  024302  032737  000010  177752          BIT     #10,@#HITMIS
5576  024310  001006                          BNE     GG7
5577  024312  013737  024516  001634          MOV     GGFLG1,$TMP1
5578  024320  010137  001636                  MOV     R1,$TMP2
5579  024324  104001                  1$:     ERROR   1
5580
5581  024326  020102                  GG7:    CMP     R1,R2           ;DOES @#ADDRESS CONTAIN
5582  024330  001412                          BEQ     GG8             ;#ADDRESS
5583
5584  024332  013737  024516  001634          MOV     GGFLG1,$TMP1
5585  024340  010137  001636                  MOV     R1,$TMP2
5586  024344  010237  001640                  MOV     R2,$TMP3
5587  024350  010137  001642                  MOV     R1,$TMP4
5588  024354  104016                  1$:     ERROR   16
5589
5590  024356  005121                  GG8:    COM     (R1)+           ;COMPLIMENT DATA
5591  024360  077032                          SOB     R0,GG6          ;LOOP FOR NEXT ADDRESS.
5592  024362  012701  140000                  MOV     #TESTR1,R1
5593  024366  012700  001000                  MOV     #1000,R0
5594  024372  012737  024400  001510          MOV     #GG9,$LPERR
5595  024400  000240                  GG9:    NOP
5596  024402  011102                          MOV     (R1),R2         ;GO BACK AND CHECK
5597  024404  032737  000010  177752          BIT     #10,@#HITMIS    ;COMPLIMENTED DATA
5598  024412  001006                          BNE     GG10
5599  024414  013737  024516  001634          MOV     GGFLG1,$TMP1
5600  024422  010137  001636                  MOV     R1,$TMP2
5601  024426  104001                  1$:     ERROR   1
5602                                          ;?????
5603
5604  024430  010103                  GG10:   MOV     R1,R3           ;IS COMPLIMENT DATA CORRECT?
5605  024432  005103                          COM     R3
5606  024434  020302                          CMP     R3,R2
5607  024436  001412                          BEQ     GG11
5608  024440  013737  024516  001634          MOV     GGFLG1,$TMP1
5609  024446  010337  001636                  MOV     R3,$TMP2
5610  024452  010237  001640                  MOV     R2,$TMP3
5611  024456  010137  001642                  MOV     R1,$TMP4
5612  024462  104016                  1$:     ERROR   16
5613
5614  024464  005721                  GG11:   TST     (R1)+           ;TEST NEXT LOCATION
5615  024466  077034                          SOB     R0,GG9
5616
5617  024470  012737  000034  024520          MOV     #SOMOM1,GGGS    ;GO BACK AND RUN
5618  024476  012737  000054  024522          MOV     #S1MOM1,GGGM    ;TEST IN GROUP 0.
5619  024504  005337  024516                  DEC     GGFLG1
5620  024510  001005                          BNE     GG12
5621  024512  000137  024070                  JMP     GG2
5622
```

```
5623  024516  000000                  GGFLG1: .WORD   0               ;GROUP BEING TESTED, 0 OR 1.
5624
5625  024520  000000                  GGGS:   .WORD   0               ;CACHE CONTPOL REGISTER
5626  024522  000000                  GGGM:   .WORD   0               ;PATTERNS
5627
5628  024524                          GG12:                           ;DONE!
5629
5630                                  ;;*****************************************************************
5631                                  ;*TEST 23       CACHE DATA MEMORY COUNT PATTERN TEST
5632                                  ;*
5633                                  ;*THIS TEST RUNS A COUNT PATTERN THROUGH EACH LOCATION
5634                                  ;*OF THE CACHE DATA MEMORY FOR EACH GROUP.
5635                                  ;*
5636                                  ;;*****************************************************************
5637  024524  000004                  TST23:  SCOPE
5638  024526  012737  000010  001702          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
5639                                                                  ;SET THE SKAD REGISTER
5640  024534  012737  025504  043632          MOV     #TST24,SKAD     ;IN CASE THE TEST ABORTS.
5641
5642  024542  012737  043500  000114          MOV     #SPUR,@#CACHVEC
5643  024550  113737  001502  001632          MOVB    $TSTNM,$TMP0
5644
5645  024556  012737  000001  025212  LL1:    MOV     #1,LLFLG1       ;TEST GROUP ONE FIRST
5646  024564  012737  000044  025220          MOV     #SIMD,LLGS      ;SIMD AND SOMI ARE PATTERNS
5647  024572  012737  000030  025222          MOV     #SOMI,LLGM      ;WHICH WILL BE LOADED INTO
5648  024600  012737  024600  001510  LL2:    MOV     #LL2,$LPERR     ;THE CACHE CONTROL REGISTER.
5649  024606  012737  043500  000114          MOV     #SPUR,@#CACHVEC
5650  024614  012700  024600                  MOV     #LL2,R0         ;MAKE THIS CODE, LOCATIONS
5651  024620  012701  001000                  MOV     #1000,R1        ;LL2 THROUGH LL2+2000 (OCT)
5652                                                                  ;HITS IN THE CACHE GROUP
5653  024624  013737  025222  177746  LL3:    MOV     LLGM,@#CONTRL   ;NOT BEING TESTED, AND MISSES
5654  024632  005710                          TST     (R0)            ;TO THE CACHE GROUP BEING
5655  024634  013737  025220  177746          MOV     LLGS,@#CONTRL   ;TESTED.
5656  024642  005760  002000                  TST     2000(R0)
5657  024646  062700  000002                  ADD     #2,R0
5658  024652  077114                          SOB     R1,LL3
5659
5660  024654  012701  140000                  MOV     #TESTR1,R1      ;MAKE THE MEMORY TEST AREA
5661  024660  012700  001000                  MOV     #1000,R0        ;HITS IN THE GROUP BEING
5662  024664  012737  024706  001510          MOV     #1$,$LPERR      ;TESTED.
5663  024672  013702  025220                  MOV     LLGS,R2
5664  024676  042702  177717                  BIC     #177717,R2
5665  024702  010237  177746                  MOV     R2,@#CONTRL
5666  024706  005011                  1$:     CLR     (R1)
5667  024710  005711                          TST     (R1)
5668  024712  005721                          TST     (R1)+
5669  024714  032737  000010  177752          BIT     #10,@#HITMIS
5670  024722  001011                          BNE     3$
5671  024724  013737  025212  001634          MOV     LLFLG1,$TMP1
5672  024732  011137  001636                  MOV     (R1),$TMP2
5673  024736  062737  177776  001636          ADD     #-2,$TMP2
5674  024744  104001                  2$:     ERROR   1
5675  024746  077021                  3$:     SOB     R0,1$
5676  024750  013700  025222                  MOV     LLGM,R0         ;FROM NOW ON SELECT
5677  024754  042700  177717                  BIC     #177717,R0      ;THE GROUP NOT BEING
5678  024760  010037  177746                  MOV     R0,@#CONTRL     ;TESTED
```

```
5679
5680  024764  012701  140000                  MOV     #TESTR1,R1      ;INITIALIZE FOR TEST.
5681  024770  012700  001000                  MOV     #1000,R0        ;COUNTER.
5682  024774  005002                  LL4:    CLR     R2              ;DATA PATTERN WRITTEN
5683  024776  005003                          CLR     R3              ;LOGICAL 'OR' OF BAD DATA
5684  025000  012704  177777                  MOV     #177777,R4      ;LOGICAL 'AND' OF BAD DATA
5685  025004  005005                          CLR     R5              ;DATA PATTERN READ
5686  025006  005037  025224                  CLR     LLCNT1          ;NUMBER OF LOCATIONS WHICH FAIL.
5687  025012  005037  025214                  CLR     LLFLG2          ;ERROR IN GROUP FLAG
5688  025016  012737  025024  001510          MOV     #LL5,$LPERR
5689  025024  005037  025216          LL5:    CLR     LLFLG4          ;ERROR IN TESTED WORD FLAG.
5690  025030  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
5691  025032  010211                          MOV     R2,(R1)
5692  025034  011105                          MOV     (R1),R5
5693  025036  032737  000010  177752          BIT     #10,@#HITMIS
5694  025044  001006                          BNE     LL6
5695  025046  013737  025212  001634          MOV     LLFLG1,$TMP1
5696  025054  010137  001636                  MOV     R1,$TMP2
5697  025060  104001                  1$:     ERROR   1
5698  025062  020205                  LL6:    CMP     R2,R5           ;GOOD DATA
5699  025064  001402                          BEQ     LL7
5700  025066  000137  025436                  JMP     LLERR2          ;BAD DATA BUT NO TRAP OR
5701                                                                  ;ABORT OCCURRED!
5702  025072                          LL7:                            ;DECREMENT THE COUNT PATTERN
5703                                                                  ;AND LOOP IF NOT DONE
5704  025072  005737  025216                  TST     LLFLG4          ;IF THERE WAS AN ERROR
5705  025076  001405                          BEQ     LL8             ;IN THE WORD JUST TESTED
5706  025100  005237  025224                  INC     LLCNT1          ;INCREMENT LLCNT1
5707  025104  012737  177777  025214          MOV     #-1,LLFLG2      ;AND SET ERROR IN GROUP FLAG.
5708  025112  062701  000002          LL8:    ADD     #2,R1           ;GO TO NEXT WORD.
5709  025116  077036                          SOB     R0,LL5
5710
5711  025120  005737  025214                  TST     LLFLG2          ;DONE WITH THAT GROUP,
5712  025124  001417                          BEQ     LL9             ;SEE IF THERE WERE
5713  025126  112737  000013  001514          MOVB    #13,$ITEMB      ;ANY ERRORS. IF SO THEN
5714  025134  013737  025212  001634          MOV     LLFLG1,$TMP1    ;PRINT AN ERROR SUMMARY
5715  025142  010437  001636                  MOV     R4,$TMP2        ;FOR THAT GROUP.
5716  025146  010337  001640                  MOV     R3,$TMP3
5717  025152  013737  025224  001642          MOV     LLCNT1,$TMP4
5718  025160  004737  044334                  JSR     PC,ERTYPE
5719
5720  025164  012737  000044  025222  LL9:    MOV     #S1M0,LLGM      ;TEST THE OTHER GROUP, 0.
5721  025172  012737  000030  025220          MOV     #S0M1,LLGS      ;
5722  025200  005337  025212                  DEC     LLFLG1
5723  025204  001137                          BNE     LL10            ;DONE?
5724  025206  000137  004600                  JMP     LL2
5725
5726  025212  000000          LLFLG1: .WORD   0               ;GROUP BEING TESTED, 1 OR 0.
5727  025214  000000          LLFLG2: .WORD   0               ;ERROR OCCURRED IN GROUP FLAG.
5728
5729  025216  000000          LLFLG4: .WORD   0               ;ERROR OCCURRED IN WORD FLAG.
5730
5731  025220  000000          LLGS:   .WORD   0               ;PATTERNS FOR CONTROL REGISTER
5732  025222  000000          LLGM:   .WORD   0
5733
5734  025224  000000          LLCNT1: .WORD   0               ;GROUP ERROR COUNT
```

```
5735
5736  025226  000000                    LLMER:  .WORD   0               ;TEMPORARY STORAGE FOR
5737                                                                    ;THE CACHE ERROR REGISTER.
5738  025230  000000                    LLTMP1: .WORD   0
5739
5740  025232  013737  177744  025226    LLERR1: MOV     @#MEMERR,LLMER  ;COME HERE ON PARITY
5741  025240  012737  004100  025230            MOV     #4100,LLTMP1    ;ABORT OR TRAP.
5742  025246  005737  025212                    TST     LLFLG1          ;TESTING GROUP 1 OR 0?
5743  025252  001403                            BEQ     1$
5744  025254  012737  004200  025230            MOV     #4200,LLTMP1
5745  025262  023737  025230  025226  1$:       CMP     LLTMP1,LLMER    ;WAS THE ERROR EXPECTED?
5746  025270  001402                            BEQ     2$
5747  025272  000137  043500                    JMP     SPUR            ;NO!
5748
5749  025276  020137  177740            2$:     CMP     R1,@#LOADRS     ;WAS THAT ADDRESS EXPECTED?
5750  025302  001402                            BEQ     3$
5751  025304  000137  043500                    JMP     SPUR            ;NO!
5752
5753  025310  012737  177777  025216  3$:       MOV     #-1,LLFLG4      ;SET WORD ERROR FLAG
5754  025316  050203                            BIS     R2,R3           ;DO 'OR' OF FAILIING DATA
5755  025320  005102                            COM     R2
5756  025322  040204                            BIC     R2,R4           ;DO 'AND' OF FAILING DATA
5757  025324  005102                            COM     R2
5758  025326  011637  001634                    MOV     (SP),$TMP1
5759  025332  022626                            CMP     (SP)+,(SP)+
5760  025334  013737  025212  001636            MOV     LLFLG1,$TMP2
5761  025342  010237  001640                    MOV     R2,$TMP3
5762  025346  010137  001650                    MOV     R1,$TMP7
5763  025352  013737  177740  001642            MOV     @#LOADRS,$TMP4
5764  025360  013737  177742  001644            MOV     @#HIADRS,$TMP5
5765  025366  042737  140000  001644            BIC     #140000,$TMP5
5766  025374  013737  025226  001646            MOV     LLMER,$TMP6
5767  025402  104011                            ERROR   11              ;REPORT ERROR.
5768
5769  025404  012737  025416  000114            MOV     #LLERR3,@#CACHVEC       ;BEFORE CONTINUING THE
5770                                                                    ;BAD PARITY IN THE WORD
5771                                                                    ;BEING TESTED MUST BE
5772                                                                    ;DEALT WITH!
5773  025412  005011                            CLR     (R1)            ;THIS INSTRUCTION CLR (R1)
5774  025414  005711                            TST     (R1)            ;SHOULD TRAP!
5775
5776  025416  012737  177777  177744  LLERR3: MOV     #-1,@#MEMERR    ;CLR THE ERROR REGISTER
5777  025424  012737  025232  000114            MOV     #LLERR1,@#CACHVEC       ;RESTORE THE PARITY ERROR
5778  025432  000137  025072                    JMP     LL7             ;VECTOR AND CONTINUE.
5779
5780  025436  012737  177777  025216  LLERR2: MOV     #-1,LLFLG4      ;BAD DATA WAS READ BUT
5781                                                                    ;NO TRAP OR ABORT OCCURRED!
5782  025444  050203                            BIS     R2,R3           ;'OR' BAD DATA
5783  025446  005102                            COM     R2
5784  025450  040204                            BIC     R2,R4           ;'AND' BAD DATA
5785  025452  005102                            COM     R2
5786  025454  013737  025212  001634            MOV     LLFLG1,$TMP1
5787  025462  010137  001640                    MOV     R1,$TMP3
5788  025466  010237  001642                    MOV     R2,$TMP4
5789  025472  010537  001644                    MOV     R5,$TMP5
5790
```

```
5791  025476  104012                  1$:     ERROR   12              ;REPORT ERROR.
5792
5793  025500  000137  025072                  JMP     LL7             ;CONTINUE TEST.
5794  025504                          LL10:
5795
5796
5797                                  ;;*****************************************************************
5798                                  ;*TEST 24       CACHE DATA MEMORY PARITY CHECKERS LOW BYTE TEST
5799                                  ;*
5800                                  ;*THIS IS A TEST OF THE TWO CACHE DATA MEMORY PARITY
5801                                  ;*CHECKERS FOR THE LOW BYTE, ONE FOR EACH GROUP. THE
5802                                  ;*MAINTENANCE REGISTER ISUSED TO FORCE A PARITY A
5803                                  ;*PARITY ERROR AT EVERY DATA PATTERN WHICH HAS A ONE
5804                                  ;*PARITY BIT. NOTE THAT THE CACHE DATA MEMORY PARITY HAS,
5805                                  ;*EFFECTIVELY, ODD PARITY. THE MAINTENANCE FUNCTION ON THE
5806                                  ;*CACHE DATA MEMORY PARITY CHECKERS HAS THE EFFECT OF
5807                                  ;*FORCING THE PARITY BIT OF THE BYTE BEING CHECKED TO
5808                                  ;*ZERO. THIS MEANS THAT ONCE THIS MAINTENANCE FUNCTION
5809                                  ;*IS ENABLED THE ERROR WILL OCCUR ON A SUBSEQUENT
5810                                  ;*READ OF A BYTE WITH A ONE PARITY BIT, THAT IS
5811                                  ;*BYTES WITH ZERO PARITY BITS WILL NOT CAUSE THE ERROR.
5812                                  ;*
5813                                  ;*****************************************************************
5814  025504  000004                  TST24:  SCOPE
5815  025506  012737  000020  001702          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
5816           000025                 IIA=$TN
5817                                                                  ;SET THE SKAD REGISTER
5818  025514  012737  026160  043632          MOV     #TST25,SKAD     ;IN CASE THE TEST ABORTS.
5819
5820  025522  113737  001502  001632          MOVB    $TSTNM,$TMP0
5821  025530  012737  043500  000114          MOV     #SPUR,@#CACHVEC
5822
5823  025536  005000                          CLR     R0              ;THIS IS THE COUNTER CONTAINING
5824                                                                  ;THE TEST DATA PATTERN
5825  025540  012737  025540  001510  IIA1:   MOV     #IIA1,$LPERR
5826  025546  004737  044072                  JSR     PC,PARCNT       ;SET IF THIS TEST PATTERN HAS
5827  025552  032702  000001                  BIT     #BIT0,R2        ;THE PARITY BIT SET (1), IF NOT
5828  025556  001402                          BEQ     IIA2            ;GO TO THE NEXT PATTERN
5829  025560  000137  026140                  JMP     IIA7
5830  025564  012737  000030  177746  IIA2:   MOV     #SOM1,@#CONTRL  ;SELECT GROUP ZERO.
5831  025572  012737  026044  000114          MOV     #IIAR1,@#CACHVEC        ;SET UP FOR THE ERROR
5832  025600  012705  026042                  MOV     #IIAT1,R5       ;MAKE THE TEST ADDRESS A
5833  025604  005715                          TST     (R5)            ;HIT IN GROUP ZERO
5834  025606  005715                          TST     (R5)            ;MAKE SURE IT IS A HIT
5835
5836                                                                  ;SEE IF REFERENCE ADDRESS
5837  025610  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
5838  025616  001007                          BNE     1$
5839                                                                  ;IF NOT ERROR!
5840  025620  010537  001636                  MOV     R5,$TMP2
5841  025624  012737  000000  001634          MOV     #0,$TMP1
5842  025632  104001                          ERROR   1
5843
5844  025634  104410                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
5845
5846
```

```
5847  025636  012704  000020          1$:     MOV     #20,R4  ;THIS PATTERN WILL BE
5848  025642  012702  177750                  MOV     #MAINT,R2       ;PUT IN THE MAINTENANCE
5849  025646  005001                          CLR     R1              ;REGISTER
5850  025650  010015                          MOV     R0,(R5)         ;PUT THE TEST PATTERN IN
5851                                                                  ;THE TEST ADDRESS
5852  025652  000402                          BR      64$
5853
5854          025654                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5855          025654                          LOC=-4&LOC
5856          025660                          LOC=LOC+4
5857          025660                          .=LOC
5858
5859                                                                  ;THE REFERENCE TO THIS NEXT INSTRUCTION
5860                                                                  ;WILL MAKE THE COMPARE INSTRUCTION A HIT
5861                                                                  ;SO THAT NO SPURIOUS ERROR SHOULD OCCUR
5862                                                                  ;WHILE THE MAINTENANCE REGISTER IS SET!
5863  025660  010412                  64$:    MOV     R4,(R2)         ;TURN ON THE MAINT. REG.
5864  025662  021500                          CMP     (R5),R0         ;THE REFERENCE TO (R5)
5865  025664  010112                          MOV     R1,(R2)         ;SHOULD CAUSE THE ERROR.
5866
5867  025666                          IIA3:
5868                                                                  ;THE ERROR DIDN'T OCCUR!
5869  025666  010037  001636                  MOV     R0,$TMP2        ;REPORT FAILURE
5870  025672  012737  026042  001640          MOV     #IIAT1,$TMP3
5871  025700  005037  001642                  CLR     $TMP4
5872  025704  104144                  64$:    ERROR   144
5873
5874  025706  012737  026104  000114  IIA4:   MOV     #IIAR2,@#CACHVEC        ;SET UP FOR THE GROUP ONE
5875  025714  012737  025706  001510          MOV     #IIA4,$LPERR    ;ERROR
5876  025722  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;SELECT GROUP ONE
5877
5878  025730  012705  026042                  MOV     #IIAT1,R5       ;MAKE THE TEST ADDRESS A
5879  025734  005715                          TST     (R5)            ;HIT, IN GROUP ONE.
5880  025736  005715                          TST     (R5)
5881
5882                                                                  ;SEE IF REFERENCE ADDRESS
5883  025740  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
5884  025746  001007                          BNE     1$
5885                                                                  ;IF NOT ERROR!
5886  025750  010537  001636                  MOV     R5,$TMP2
5887  025754  012737  000001  001634          MOV     #1,$TMP1
5888  025762  104001                          ERROR   1
5889
5890  025764  104410                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
5891
5892
5893  025766  012704  000100          1$:     MOV     #100,R4 ;THIS PATTERN WILL BE
5894  025772  012702  177750                  MOV     #MAINT,R2       ;PUT IN THE MAINT. REG.
5895  025776  005001                          CLR     R1
5896  026000  010015                          MOV     R0,(R5)         ;PUT THE TEST PATTERN IN (R5),
5897                                                                  ;IIAT1.
5898  026002  000402                          BR      50$             ;PUT THE NEXT INSTRUCTION EXECUTED
5899                                                                  ;ON AN EVEN WORD BOUNDARY SO THE
5900                                                                  ;SUBSEQUENT INSTRUCTION, A CMP,
5901                                                                  ;WILL BE A HIT.
5902
```

```
                026004                          LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                026004                          LOC=-4&LOC
                026010                          LOC=LOC+4
                026010                          .=LOC

026010  000240                  50$:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
026012  010412                          MOV     R4,(R2)         ;TURN ON THE MAINT. REG.
026014  021500                          CMP     (R5),R0         ;THIS REFERENCE TO (R5) SHOULD
026016  010112                          MOV     R1,(R2)         ;CAUSE THE ERROR.

026020                          IIA5:
                                                                ;THE ERROR DIDN'T OCCUR!
026020  010037  001636                  MOV     R0,$TMP2        ;REPORT FAILURE
026024  012737  026042  001640          MOV     #IIAT1,$TMP3
026032  005037  001642                  CLR     $TMP4
026036  104145                  64$:    ERROR   145

026040  000437                  IIA6:   BR      IIA7

026042  000000                  IIAT1:.WORD     0

026044                          IIAR1:
026044  022737  004500  177744          CMP     #4500,@#MEMERR  ;MAKE SURE THE ERROR
026052  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
026054  000137  043500          1$:     JMP     SPUR
026060  022737  026042  177740  2$:     CMP     #IIAT1,@#LOADRS ;MAKE SURE THE ERROR
026066  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
026070  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
026072  012737  177777  177744          MOV     #-1,@#MEMEPR    ;CLEAR THE ERROR REGISTERS.
026100  000137  025706                  JMP     IIA4            ;GO TEST GROUP ONE
026104                          IIAR2:
026104  022737  004600  177744          CMP     #4600,@#MEMERR  ;MAKE SURE THE ERROR
026112  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
026114  000137  043500          1$:     JMP     SPUR
026120  022737  026042  177740  2$:     CMP     #IIAT1,@#LOADRS ;MAKE SURE THE ERROR
026126  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
026130  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
026132  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.

026140  022700  000377          IIA7:   CMP     #377,R0         ;INCREMENT THE TEST
026144  001404                          BEQ     IIA8            ;PATTERN
026146  062700  000001                  ADD     #1,R0
026152  000137  025540                  JMP     IIA1

026156  104407                  IIA8:   RSET

                                ;;*****************************************************************
                                ;*TEST 25        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.
                                        ;*
                                        ;**************************************************************
026160  000004                TST25:  SCOPE
026162  012737  000020  001702        MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000026          IIB=$TN
                                                                ;SET THE SKAD REGISTER
026170  012737  026634  043632        MOV     #TST26,SKAD     ;IN CASE THE TEST ABORTS.

026176  113737  001502  001632        MOVB    $TSTNM,$TMPO
026204  012737  043500  000114        MOV     #SPUR,@#CACHVEC

026212  005000                        CLR     R0              ;THIS IS THE COUNTER CONTAINING
                                                                ;THE TEST DATA PATTERN
026214  012737  026214  001510  IIB1:  MOV     #IIB1,$LPERR
026222  004737  044072                JSR     PC,PARCNT       ;SET IF THIS TEST PATTERN HAS
026226  032702  000001                BIT     #BIT0,R2        ;THE PARITY BIT SET (1), IF NOT
026232  001402                        BEQ     IIB2            ;GO TO THE NEXT PATTERN
026234  000137  026614                JMP     IIB7
026240  012737  000030  177746  IIB2:  MOV     #SOM1,@#CONTRL  ;SELECT GROUP ZERO.
026246  012737  026520  000114        MOV     #IIBR1,@#CACHVEC        ;SET UP FOR THE ERROR
026254  012705  026516                MOV     #IIBT1,R5       ;MAKE THE TEST ADDRESS A
026260  005715                        TST     (R5)            ;HIT IN GROUP ZERO
026262  005715                        TST     (R5)            ;MAKE SURE IT IS A HIT

                                                                ;SEE IF REFERENCE ADDRESS
026264  032737  000010  177752        BIT     #10,@#HITMIS    ;IS A HIT.
026272  001007                        BNE     1$
                                                                ;IF NOT ERROR!
026274  010537  001636                MOV     R5,$TMP2
026300  012737  000000  001634        MOV     #0,$TMP1
026306  104001                        ERROR   1

026310  104410                        SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

026312  012704  000040          1$:   MOV     #40,R4  ;THIS PATTERN WILL BE
026316  012702  177750                MOV     #MAINT,R2       ;PUT IN THE MAINTENANCE
026322  005001                        CLR     R1              ;REGISTER
026324  010015                        MOV     R0,(R5)         ;PUT THE TEST PATTERN IN
                                                                ;THE TEST ADDRESS
026326  000402                        BR      64$

        026330                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        026330                  LOC=-4&LOC
        026334                  LOC=LOC+4
        026334                  .=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!
026334  010412                  64$:    MOV     R4,(R2)                 ;TURN ON THE MAINT. REG.
026336  021500                          CMP     (R5),R0                 ;THE REFERENCE TO (R5)
026340  010112                          MOV     R1,(R2)                 ;SHOULD CAUSE THE ERROR.

026342                          IIB3:
                                                                        ;THE ERROR DIDN'T OCCUR!
026342  010037  001636                  MOV     R0,$TMP2                ;REPORT FAILURE
026346  012737  026516  001640          MOV     #IIBT1,$TMP3
026354  005037  001642                  CLR     $TMP4
026360  104146                  64$:    ERROR   146

026362  012737  026560  000114  IIB4:   MOV     #IIBR2,@#CACHVEC        ;SET UP FOR THE GROUP ONE
026370  012737  026362  001510          MOV     #IIB4,$LPERR    ;ERROR
026376  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;SELECT GROUP ONE

026404  012705  026516                  MOV     #IIBT1,R5               ;MAKE THE TEST ADDRESS A
026410  005715                          TST     (R5)                    ;HIT, IN GROUP ONE.
026412  005715                          TST     (R5)

                                                                        ;SEE IF REFERENCE ADDRESS
026414  032737  000010  177752          BIT     #10,@#HITMIS            ;IS A HIT.
026422  001007                          BNE     1$

                                                                        ;IF NOT ERROR!
026424  010537  001636                  MOV     R5,$TMP2
026430  012737  000001  001634          MOV     #1,$TMP1
026436  104001                          ERROR   1

026440  104410                          SKIPT                           ;ERROR FATAL. GO TO NEXT TEST.


026442  012704  000200          1$:     MOV     #200,R4 ;THIS PATTERN WILL BE
026446  012702  177750                  MOV     #MAINT,R2               ;PUT IN THE MAINT. REG.
026452  005001                          CLR     R1
026454  010015                          MOV     R0,(R5)                 ;PUT THE TEST PATTERN IN (R5),
                                                                        ;IIBT1.
026456  000402                          BR      50$                     ;PUT THE NEXT INSTRUCTION EXECUTED
                                                                        ;ON AN EVEN WORD BOUNDARY SO THE
                                                                        ;SUBSEQUENT INSTRUCTION, A CMP,
                                                                        ;WILL BE A HIT.

        026460                          LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        026460                          LOC=-4&LOC
        026464                          LOC=LOC+4
        026464                          .=LOC

026464  000240                  50$:    NOP                             ;FOR SCOPING WITH AN OSCILLOSCOPE.
026466  010412                          MOV     R4,(R2)                 ;TURN ON THE MAINT. REG.
026470  021500                          CMP     (R5),R0                 ;THIS REFERENCE TO (R5) SHOULD
026472  010112                          MOV     R1,(R2)                 ;CAUSE THE ERROR.

026474                          IIB5:
                                                                        ;THE ERROR DIDN'T OCCUR!
026474  010037  001636                  MOV     R0,$TMP2                ;REPORT FAILURE
026500  012737  026516  001640          MOV     #IIBT1,$TMP3
```

```
 6071  026506  005037  001642                CLR     $TMP4
 6072  026512  104147                64$:    ERROR   147
 6073
 6074  026514  000437                IIB6:   BR      IIB7
 6075
 6076  026516  000000                IIBT1:.WORD     0
 6077
 6078  026520                        IIBR1:
 6079  026520  022737  004500  177744        CMP     #4500,@#MEMERR  ;MAKE SURE THE ERROR
 6080  026526  001402                        BEQ     2$              ;REGISTER IS SET PROPERLY
 6081  026530  000137  043500        1$:     JMP     SPUR
 6082  026534  022737  026516  177740  2$:   CMP     #IIBT1,@#LOADRS ;MAKE SURE THE ERROR
 6083  026542  001372                        BNE     1$              ;OCCURRED AT THE CORRECT
 6084                                                                ;ADDRESS.
 6085  026544  022626                        CMP     (SP)+,(SP)+     ;RESET THE STACK
 6086  026546  012737  177777  177744        MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
 6087  026554  000137  026362                JMP     IIB4            ;GO TEST GROUP ONE
 6088  026560                        IIBR2:
 6089  026560  022737  004600  177744        CMP     #4600,@#MEMERR  ;MAKE SURE THE ERROR
 6090  026556  001402                        BEQ     2$              ;REGISTER IS SET PROPERLY
 6091  026570  000137  043500        1$:     JMP     SPUR
 6092  026574  022737  026516  177740  2$:   CMP     #IIBT1,@#LOADRS ;MAKE SURE THE ERROR
 6093  026602  001372                        BNE     1$              ;OCCURRED AT THE CORRECT
 6094                                                                ;ADDRESS.
 6095  026604  022626                        CMP     (SP)+,(SP)+     ;RESET THE STACK
 6096  026606  012737  177777  177744        MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
 6097
 6098  026614  022700  177400        IIB7:   CMP     #177400,R0              ;INCREMENT THE TEST
 6099  026620  001404                        BEQ     IIB8            ;PATTERN
 6100  026622  062700  000400                ADD     #400,R0
 6101  026626  000137  026214                JMP     IIB1
 6102
 6103  026632  104407                IIB8:   RSET
 6104
 6105
 6106                                ;;*****************************************************************
 6107                                ;*TEST 26       CACHE DATA MEMORY WORST CASE NOISE TEST
 6108                                ;*
 6109                                ;*THIS TEST DOES A GALLOPING 0'S AND 1'S OR PING PONG
 6110                                ;*TEST ON THE CACHE BIPOLAR DATA MEMORY.
 6111                                ;*
 6112                                ;;*****************************************************************
 6113  026634  000004                TST26:  SCOPE
 6114                                                                ;SET THE SKAD REGISTER
 6115  026636  012737  027772  043632        MOV     #TST27,SKAD     ;IN CASE THE TEST ABORTS.
 6116
 6117
 6118  026644  012737  043500  000114        MOV     #SPUR,@#CACHVEC
 6119  026652  113737  001502  001632        MOVB    $TSTNM,$TMP0    ;SAVE TESTN FOR PRINT OUT.
 6120
 6121  026660  005037  027362                CLR     QQPAT1          ;BACK ROUND PATTERN OF
 6122                                                                ;0'S FOR THE GALLOPING
 6123                                                                ;1'S TEST TO BE EXECUTED
 6124                                                                ;FIRST.
 6125  026664  012737  000001  027356        MOV     #1,QQFLG2       ;QQFLG=1 MEANS GALLOPING
 6126                                                                ;ONES TEST IN PROGRESS.
```

```
6127                                                                    ;QQFLG=0 MEANS GALLOPING
6128                                                                    ;ZEROES TEST IN PROGRESS.
6129  026672  012737  026672  001510  QQ1:    MOV     #QQ1,$LPERR     ;SET ERROR LOOP INITIALLY
6130                                                                    ;TO THIS POINT.
6131  026700  012737  000044  027372          MOV     #S1M0,QQGS      ;TEST GROUP 1 FIRST.
6132  026706  012737  000030  027374          MOV     #S0M1,QQGM      ;S0M1 AND S1M0 ARE
6133                                                                    ;PATTERNS WHICH WILL BE
6134                                                                    ;LOADED INTO THE CACHE
6135                                                                    ;CONTROL REGISTER TO
6136                                                                    ;(SELECT GRP0 * MISS GRP1)
6137                                                                    ;AND (SELECT GRP1 * MISS GRP0)
6138                                                                    ;RESPECTIVELY.
6139  026714  012737  000001  027360          MOV     #1,QQFLG1       ;QQFLG ONE CONTAINS THE
6140                                                                    ;NUMBER OF THE GROUP
6141                                                                    ;BEING TESTED, INITIALLY 1.
6142
6143  026722  012703  026722          QQ2:    MOV     #QQ2,R3         ;MAKE LOCATIONS QQ1
6144  026726  012704  001000                  MOV     #1000,R4        ;THROUGH QQ2 + 2000 (OCT)
6145  026732  013737  027374  177746  1$:     MOV     QQGM,@#CONTRL   ;HITS IN THE GROUP NOT
6146  026740  005713                          TST     (R3)            ;BEING TESTED WHILE
6147  026742  013737  027372  177746          MOV     QQGS,@#CONTRL   ;GETTING THESE LOCATIONS
6148  026750  005763  002000                  TST     2000(R3)        ;TO BE MISSES IN THE
6149  026754  062703  000002                  ADD     #2,R3           ;GROUP THAT IS BEING
6150  026760  077414                          SOB     R4,1$           ;TESTED
6151  026762  012704  001000                  MOV     #1000,R4        ;MAKE LOCATIONS TESTR2
6152  026766  012705  142000                  MOV     #TESTR2,R5      ;THROUGH TESTR2+2000(OCT)
6153  026772  013703  027372                  MOV     QQGS,R3         ;HITS IN THE GROUP
6154  026776  042703  177717                  BIC     #177717,R3      ;BEING TESTED WHILE
6155  027002  010337  177746                  MOV     R3,@#CONTRL     ;WRITING THE BACKGROUND
6156  027006  013715  027362          QQ3:    MOV     QQPAT1,(R5)     ;PATTERN, IN QQPAT1, IN
6157  027012  005715                          TST     (R5)
6158  027014  005725                          TST     (R5)+           ;THEM. MAKE SURE THEY
6159  027016  032737  000010  177752          BIT     #10,@#HITMIS    ;ARE HITS
6160  027024  001011                          BNE     QQ4
6161  027026  013737  027360  001634          MOV     QQFLG1,$TMP1    ;IF NOT ERROR
6162  027034  010537  001636                  MOV     R5,$TMP2
6163  027040  062737  177776  001636          ADD     #-2,$TMP2
6164  027046  104001                  1$:     ERROR   1
6165  027050  077422                  QQ4:    SOB     R4,QQ3
6166  027052  013703  027374                  MOV     QQGM,R3         ;FROM NOW ON SELECT
6167  027056  042703  177717                  BIC     #177717,R3      ;THE GROUP NOT BEING
6168  027062  010337  177746                  MOV     R3,@#CONTRL     ;TESTED
6169
6170  027066  012704  027376                  MOV     #QQ10,R4        ;THE THREE ROUTINES
6171  027072  042704  176000                  BIC     #176000,R4      ;QQ10-QQ11, QQ12-QQ13 AND
6172  027076  012705  027452                  MOV     #QQ11,R5        ;QQ14-QQ15 ARE IDENTICAL
6173  027102  042705  176000                  BIC     #176000,R5      ;EXCEPT FOR WHAT PART
6174  027106  020405                          CMP     R4,R5           ;OF THE CACHE GROUP THAT
6175  027110  002407                          BLT     QQ5             ;IS NOT BEING TEST THEY
6176  027112  012737  027454  027352          MOV     #QQ12,QQLO      ;LIE IN. HERE DECIDE
6177  027120  012737  027532  027354          MOV     #QQ14,QQHI      ;WHICH TWO OF THE
6178  027126  000450                          BR      QQ8             ;ABOVE THREE IS APPROPRIATE
6179  027130  012704  027454          QQ5:    MOV     #QQ12,R4        ;FOR THIS TEST.
6180  027134  042704  176000                  BIC     #176000,R4
6181  027140  012705  027530                  MOV     #QQ13,R5
6182  027144  042705  176000                  BIC     #176000,R5
```

```
6183  027150  020405                          CMP     R4,R5
6184  027152  002407                          BLT     QQ6
6185  027154  013737  027532  027352          MOV     QQ14,QQLO
6186  027162  013737  027376  027354          MOV     QQ10,QQHI
6187  027170  000427                          BR      QQ8
6188  027172  013704  027376          QQ6:    MOV     QQ10,R4
6189  027176  042704  176000                  BIC     #176000,R4
6190  027202  012705  027454                  MOV     #QQ12,R5
6191  027206  042705  176000                  BIC     #176000,R5
6192  027212  020405                          CMP     R4,R5
6193  027214  003007                          BGT     QQ7
6194  027216  012737  027376  027352          MOV     #QQ10,QQLO
6195  027224  012737  027454  027354          MOV     #QQ12,QQHI
6196  027232  000406                          BR      QQ8
6197  027234  012737  027454  027352  QQ7:    MOV     #QQ12,QQLO
6198  027242  012737  027376  027354          MOV     #QQ10,QQHI
6199
6200  027250  012702  142000          QQ8:    MOV     #TESTR2,R2      ;INITIALIZE FOR EITHER
6201  027254  012701  140000                  MOV     #TESTR1,R1      ;THE GALLOPING ONES OR
6202  027260  012705  001000                  MOV     #1000,R5        ;GALLOPING ZEROES TEST
6203                                                                  ;WHICH IS PENDING.
6204  027264  012737  027674  000114          MOV     #QQERR1,@#CACHVEC       ;IF THE TEST FAILS A
6205                                                                  ;PARITY ABORT IS LIKELY
6206                                                                  ;SO SET UP TO GO THE
6207                                                                  ;ERROR ROUTINE.
6208  027272  012737  027300  001510          MOV     #QQ9,$LPERR     ;SET THE LOOP ERROR
6209                                                                  ;ADDRESS FOR THE BEGINNING
6210                                                                  ;OF THE PASS ROUTINE.
6211
6212  027300  012703  142000          QQ9:    MOV     #TESTR2,R3      ;THIS DOES ONE PASS OF
6213  027304  012704  001000                  MOV     #1000,R4        ;THE TEST FOR EACH LOCATION.
6214  027310  005112                          COM     (R2)            ;PUT THE GALLOPING PATTERN
6215                                                                  ;IN THE MEMORY.
6216
6217  027312  010100          QQ9.5:  MOV     R1,R0           ;SEE WHICH OF THE
6218  027314  042700  176000                  BIC     #176000,R0      ;TWO ROUTINES (QQ10,QQ12 OR
6219  027320  013737  027354  027364          MOV     QQHI,QQTMP1     ;QQ14) SHOULD FINISH
6220  027326  042737  176000  027364          BIC     #176000,QQTMP1  ;SETTING FOR THIS TEST
6221  027334  020037  027364                  CMP     R0,QQTMP1       ;PASS.
6222  027340  002402                          BLT     1$
6223  027342  000177  000004                  JMP     @QQLO
6224  027346  000177  000002          1$:     JMP     @QQHI
6225
6226  027352  000000          QQLO:   .WORD   0               ;QQLO AND QQHI CONTAIN THE
6227  027354  000000          QQHI:   .WORD   0               ;ADDRESSES OF THE ROUTINES
6228                                                          ;TO BE USED IN SETTING UP
6229                                                          ;FOR A PASS.
6230  027356  000000          QQFLG2: .WORD   0               ;1 IF DOING GALLOPING 1'S TEST.
6231                                                          ;0 IF DOING GALLOPING 0'S TEST.
6232  027360  000000          QQFLG1: .WORD   0               ;GROUP BEING TESTED, 1 OR 0.
6233  027362  000000          QQPAT1: .WORD   0               ;0 OR 1 BACKGROUND PATTERN.
6234  027364  000000          QQTMP1: .WORD   0               ;USED AS TEMPORARY STORAGE.
6235  027366  000000          QQTMP2: .WORD   0
6236  027370  000000          QQTMP3: .WORD   0
6237  027372  000000          QQGS:   .WORD   0               ;THESE REGISTERS HOLD PATTERNS
6238  027374  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.
027376  000240                  QQ10:   NOP                     ;USED AS A BUFFER SO
027400  000240                          NOP                     ;THIS CODE WON'T WIPE
                                                                ;OUT DESIRED HITS
027402  012711  022312                  MOV     #022312,(R1)    ;020323=(CMP (R3)+,(R2)
027406  005711                          TST     (R1)
027410  012761  077402  000002          MOV     #077402,2(R1)   ;077402=(SOB R4,.-2)
027416  005761  000002                  TST     2(R1)
027422  012761  000137  000004          MOV     #000137,4(R1)   ;000137=(JMP @#QQ16)
027430  005761  000004                  TST     4(R1)           ;QQ16
027434  012761  027610  000006          MOV     #QQ16,6(R1)
027442  005761  000006                  TST     6(R1)
027446  000111                          JMP     (R1)            ;GO DO A PASS.
027450  000240                          NOP
027452  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.
027454  000240                  QQ12:   NOP                     ;USED AS A BUFFER SO
027456  000240                          NOP                     ;THIS CODE WON'T WIPE
                                                                ;OUT DESIRED HITS
027460  012711  022312                  MOV     #022312,(R1)    ;020323=(CMP (R3)+,(R2)
027464  005711                          TST     (R1)
027466  012761  077402  000002          MOV     #077402,2(R1)   ;077402=(SOB R4,.-2)
027474  005761  000002                  TST     2(R1)
027500  012761  000137  000004          MOV     #000137,4(R1)   ;000137=(JMP @#QQ16)
027506  005761  000004                  TST     4(R1)           ;QQ16
027512  012761  027610  000006          MOV     #QQ16,6(R1)
027520  005761  000006                  TST     6(R1)
027524  000111                          JMP     (R1)            ;GO DO A PASS.
027526  000240                          NOP
027530  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.
027532  000240                  QQ14:   NOP                     ;USED AS A BUFFER SO
027534  000240                          NOP                     ;THIS CODE WON'T WIPE
                                                                ;OUT DESIRED HITS
027536  012711  022312                  MOV     #022312,(R1)    ;020323=(CMP (R3)+,(R2)
027542  005711                          TST     (R1)
027544  012761  077402  000002          MOV     #077402,2(R1)   ;077402=(SOB R4,.-2)
027552  005761  000002                  TST     2(R1)
027556  012761  000137  000004          MOV     #000137,4(R1)   ;000137=(JMP @#QQ16)
027564  005761  000004                  TST     4(R1)           ;QQ16
```

```
6295  027570  012761  027610  000006          MOV     #QQ16,6(R1)
6296  027576  005761  000006                  TST     6(R1)
6297  027602  000111                          JMP     (R1)            ;GO DO A PASS.
6298  027604  000240                          NOP
6299  027606  000240                  QQ15:   NOP
6300
6301  027610  005122                  QQ16:   COM     (R2)+           ;PASS DONE. RESTORE THE
6302                                                                  ;BACKGROUND PATTERN.
6303
6304  027612  062701  000002          QQ17:   ADD     #2,R1           ;GO TO NEXT LOCATION FOR
6305                                                                  ;NEXT PASS.
6306  027616  005305                          DEC     R5              ;DO ANOTHER PASS?
6307  027620  001402                          BEQ     1$
6308  027622  000137  027300                  JMP     QQ9
6309  027626                          1$:
6310  027626  012737  000044  027374          MOV     #S1M0,QQGM      ;TESTED GROUP 1 NOW GO BACK
6311  027634  012737  000030  027372          MOV     #S0M1,QQGS      ;AND TEST GROUP 0
6312  027642  005337  027360                  DEC     QQFLG1
6313  027646  001002                          BNE     QQ18
6314  027650  000137  026722                  JMP     QQ2
6315
6316  027654  012737  177777  027362  QQ18:   MOV     #-1,QQPAT1      ;GALLOPING 1'S TEST IS
6317  027662  005337  027356                  DEC     QQFLG2          ;COMPLETE, ON BOTH GROUPS,
6318  027666  001041                          BNE     QQ19            ;SET THE BACKGROUND PATTERN
6319  027670  000137  026672                  JMP     QQ1             ;FOR GALLOPING 0'S AND GO
6320                                                                  ;BACK TO PERFORM THIS TEST
6321                                                                  ;ON BOTH GROUPS.
6322
6323  027674  013737  177744  001634  QQERR1: MOV     @#MEMERR,$TMP1  ;COME HERE IF DURING THE
6324  027702  013737  177740  001636          MOV     @#LOADRS,$TMP2  ;TEST A TRAP OR ABORT
6325  027710  013737  177742  001640          MOV     @#HIADRS,$TMP3  ;OCCURRED TO CACHVEC
6326  027716  011637  001642                  MOV     (SP),$TMP4
6327  027722  022626                          CMP     (SP)+,(SP)+
6328  027724  010137  001644                  MOV     R1,$TMP5
6329  027730  013737  027360  001646          MOV     QQFLG1,$TMP6
6330  027736  032737  000600  001634          BIT     #600,$TMP1
6331  027744  001002                          BNE     QQERR2
6332  027746  104002                          ERROR   2
6333  027750  000406                          BR      QQERR4
6334  027752  005737  027362          QQERR2: TST     QQPAT1          ;GALLOPING 1' OR 0'S?
6335  027756  001002                          BNE     QQERR3
6336  027760  104003                          ERROR   3               ;0'S.
6337  027762  000401                          BR      QQERR4
6338  027764  104004                  QQERR3: ERROR   4               ;1'S
6339  027766  000137  027610          QQERR4: JMP     QQ16            ;CONTINUE?
6340
6341  027772                          QQ19:                           ;DONE! PERHAPS PRINT SUMMARY.
6342                                                                  ;?????
6343
6344                                  ;;*****************************************************************
6345                                  ;*TEST 27       CACHE DATA MEMORY CHIP SELECTION LOGIC TEST
6346                                  ;*
6347                                  ;*THIS ROUTINE TESTS THE 'CHIP-SET' ENABLE LOGIC FOR THE CACHE DATA
6348                                  ;*MEMORY.  TO DEFINE THE TERM 'CHIP-SET' CONSIDER THE CACHE MEMORY AS
6349                                  ;*BEING DIVIDED INTO FOUR SETS OF 256 (DEC) X 1 BIT BIPOLAR MEMORY
6350                                  ;*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).
                                        ;*
                                        ;;*****************************************************************
027772  000004                        TST27:  SCOPE
027774  012737  000040  001702                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                                                        ;SET THE SKAD REGISTER
030002  012737  031536  043632                MOV     #TST30,SKAD     ;IN CASE THE TEST ABORTS.

030010  113737  001502  001632                MOVB    $TSTNM,$TMP0    ;PUT THE TEST NUMBER IN
                                                                        ;$TMP0 FOR PRINT OUT.
030016  012737  043500  000114                MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.

030024  012737  000014  177746  KK1:    MOV     #MOM1,@#CONTRL  ;FORCE MISSES AND
030032  005037  031372                CLR     KKPAT1          ;INITIALIZE THE TEST PATTERN
030036  012737  177777  031374        MOV     #177777,KKPAT2  ;TABLE
030044  012737  125252  031376        MOV     #125252,KKPAT3
030052  012737  052525  031400        MOV     #52525,KKPAT4

030060  005037  031356                CLR     KKFLG1          ;INITIALIZE KKFLG1:
                                                                ;0 MEANS WRITE PATTERNS IN
                                                                ;IN THE UPWARD DIRECTION
                                                                ;1 MEANS WRITE PATTERNS IN
                                                                ;THE DOWNWARD DIRECTION

030064  012700  031406        KK2:    MOV     #KKTMP2,R0      ;ESTABLISH AN OFFSET FOR
030070  042700  176003                BIC     #176003,R0      ;A TEST AREA WHOSE HITS
                                                                ;WILL NOT BE INTERFERRED WITH BY
030074  010001                        MOV     R0,R1           ;THE CYCLES CAUSED WHILE
030076  062701  140000                ADD     #TESTR1,R1      ;FETCHING THE TEST CODE.
030102  010002                        MOV     R0,R2
030104  062702  142000                ADD     #TESTR2,R2

030110  010137  001644                MOV     R1,$TMP5        ;SAVE THE ADDRESSES OF
030114  010137  001646                MOV     R1,$TMP6        ;THE FOUR TEST WORD LOCATIONS,
030120  062737  000002  001646        ADD     #2,$TMP6        ;FOR TYPE OUT IN CASE
030126  010237  001650                MOV     R2,$TMP7        ;OF ERROR.
030132  010237  001652                MOV     R2,$TMP10
030136  062737  000002  001652        ADD     #2,$TMP10

030144  012705  031374                MOV     #KKPAT2,R5      ;A POINTER USED IN GENERATING
                                                                ;EVERY PERMUTATION OF THE TEST
                                                                ;PATTERNS.
030150  012700  000006                MOV     #6,R0           ;R0 AND KKCNT1 ARE ALSO USED
```

```
6407  030154  012737  000004  031370          MOV     #4,KKCNT1       ;IN GENERATING THE PERMUTATIONS.
6408
6409  030162  012737  030170  001510          MOV     #KK3,$LPERR     ;WHEN LOOPING ON ERROR GO TO KK3.
6410  030170  000240                  KK3:    NOP                     ;FOR SCOPING PER POSES
6411  030172  012737  000034  177746          MOV     #SOMOM1,@#CONTRL;MAKE THE TEST AREA HITS
6412  030200  005711                          TST     (R1)            ;IN THE CACHE GROUPS.
6413  030202  005761  000002                  TST     2(R1)
6414  030206  012737  000054  177746          MOV     #S1MOM1,@#CONTRL
6415  030214  005712                          TST     (R2)
6416  030216  005762  000002                  TST     2(R2)
6417  030222  005037  177746                  CLR     @#CONTRL
6418
6419
6420  030226  005711                          TST     (R1)
6421
6422                                                                  ;SEE IF REFERENCE ADDRESS
6423  030230  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6424  030236  001006                          BNE     1$
6425                                                                  ;IF NOT ERROR!
6426  030240  010137  001636                  MOV     R1,$TMP2
6427  030244  012737  000000  001634          MOV     #0,$TMP1
6428  030252  104001                          ERROR   1
6429
6430
6431
6432  030254                          1$:
6433
6434  030254  005761  000002                  TST     2(R1)
6435
6436                                                                  ;SEE IF REFERENCE ADDRESS
6437  030260  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6438  030266  001011                          BNE     2$
6439                                                                  ;IF NOT ERROR!
6440  030270  010137  001636                  MOV     R1,$TMP2
6441  030274  062737  000002  001636          ADD     #2,$TMP2
6442  030302  012737  000000  001634          MOV     #0,$TMP1
6443  030310  104001                          ERROR   1
6444
6445
6446
6447  030312                          2$:
6448
6449  030312  005712                          TST     (R2)
6450
6451                                                                  ;SEE IF REFERENCE ADDRESS
6452  030314  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6453  030322  001006                          BNE     3$
6454                                                                  ;IF NOT ERROR!
6455  030324  010237  001636                  MOV     R2,$TMP2
6456  030330  012737  000001  001634          MOV     #1,$TMP1
6457  030336  104001                          ERROR   1
6458
6459
6460
6461  030340                          3$:
6462
```

```
6463  030340  005762  000002          TST     2(R2)
6464
6465                                                    ;SEE IF REFERENCE ADDRESS
6466  030344  032737  000010  177752  BIT     #10,@#HITMIS    ;IS A HIT.
6467  030352  001011          BNE     4$
6468                                                    ;IF NOT ERROR!
6469  030354  010237  001636  MOV     R2,$TMP2
6470  030360  062737  000002  001636  ADD     #2,$TMP2
6471  030366  012737  000001  001634  MOV     #1,$TMP1
6472  030374  104001          ERROR   1
6473
6474
6475
6476
6477  030376  005737  031366  4$:     TST     KKFLG1          ;SEE IF THE TST PATTERN
6478                                                    ;SHOULD BE WRITTEN UPWARD
6479                                                    ;OR DOWNWARD.
6480  030402  001045          BNE     KK4             ;BRANCH IF DOWNWARD
6481                                                    ;OTHERWISE WRITE IT IN THE
6482                                                    ;UPWARD DIRECTION.
6483  030404  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6484  030412  013703  031372          MOV     KKPAT1,R3               ;LOCATION KKPAT1, INTO THE
6485  030416  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 0
6486  030422  010361  000000          MOV     R3,0(R1)
6487  030426  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6488  030434  013703  031374          MOV     KKPAT2,R3               ;LOCATION KKPAT2, INTO THE
6489  030440  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 2
6490  030444  010361  000002          MOV     R3,2(R1)
6491  030450  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6492  030456  013703  031376          MOV     KKPAT3,R3               ;LOCATION KKPAT3, INTO THE
6493  030462  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 0
6494  030466  010362  000000          MOV     R3,0(R2)
6495  030472  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6496  030500  013703  031400          MOV     KKPAT4,R3               ;LOCATION KKPAT4, INTO THE
6497  030504  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 2
6498  030510  010362  000002          MOV     R3,2(R2)
6499  030514  000444          BR      KK5
6500  030516                  KK4:                    ;WRITE THE PATTERN IN THE
6501                                                  ;DOWNWARD DIRECTION
6502  030516  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6503  030524  013703  031400          MOV     KKPAT4,R3               ;LOCATION KKPAT4, INTO THE
6504  030530  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 2
6505  030534  010362  000002          MOV     R3,2(R2)
6506  030540  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6507  030546  013703  031376          MOV     KKPAT3,R3               ;LOCATION KKPAT3, INTO THE
6508  030552  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 0
6509  030556  010362  000000          MOV     R3,0(R2)
6510  030562  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6511  030570  013703  031374          MOV     KKPAT2,R3               ;LOCATION KKPAT2, INTO THE
6512  030574  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 2
6513  030600  010361  000002          MOV     R3,2(R1)
6514  030604  012737  000014  177746  MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
6515  030612  013703  031372          MOV     KKPAT1,R3               ;LOCATION KKPAT1, INTO THE
6516  030616  005037  177746          CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 0
6517  030622  010361  000000          MOV     R3,0(R1)
6518
```

```
6519  030626                          KK5:
6520  030626  012737  000014  177746          MOV     #MOM1,@#CONTRL
6521  030634  013703  031372                  MOV     KKPAT1,R3               ;SEE IF THE TEST PATTERN WAS
6522  030640  005037  177746                  CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
6523  030644  016104  000000                  MOV     0(R1),R4
6524
6525                                                                  ;SEE IF REFERENCE ADDRESS
6526  030650  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6527  030656  001006                          BNE     64$
6528                                                                  ;IF NOT ERROR!
6529  030660  010137  001636                  MOV     R1,$TMP2
6530  030664  012737  000000  001634          MOV     #0,$TMP1
6531  030672  104001                          ERROR   1
6532
6533
6534  030674  020403                  64$:    CMP     R4,R3
6535  030676  001402                          BEQ     KK6
6536  030700  004737  031416                  JSR     PC,KKERR1
6537
6538  030704                          KK6:
6539  030704  012737  000014  177746          MOV     #MOM1,@#CONTRL
6540  030712  013703  031374                  MOV     KKPAT2,R3               ;SEE IF THE TEST PATTERN WAS
6541  030716  005037  177746                  CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
6542  030722  016104  000002                  MOV     2(R1),R4
6543
6544                                                                  ;SEE IF REFERENCE ADDRESS
6545  030726  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6546  030734  001011                          BNE     64$
6547                                                                  ;IF NOT ERROR!
6548  030736  010137  001636                  MOV     R1,$TMP2
6549  030742  062737  000002  001636          ADD     #2,$TMP2
6550  030750  012737  000000  001634          MOV     #0,$TMP1
6551  030756  104001                          ERROR   1
6552
6553
6554  030760  020403                  64$:    CMP     R4,R3
6555  030762  001402                          BEQ     KK7
6556  030764  004737  031430                  JSR     PC,KKERR2
6557
6558  030770                          KK7:
6559  030770  012737  000014  177746          MOV     #MOM1,@#CONTRL
6560  030776  013703  031376                  MOV     KKPAT3,R3               ;SEE IF THE TEST PATTERN WAS
6561  031002  005037  177746                  CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
6562  031006  016204  000000                  MOV     0(R2),R4
6563
6564                                                                  ;SEE IF REFERENCE ADDRESS
6565  031012  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6566  031020  001006                          BNE     64$
6567                                                                  ;IF NOT ERROR!
6568  031022  010237  001636                  MOV     R2,$TMP2
6569  031026  012737  000001  001634          MOV     #1,$TMP1
6570  031034  104001                          ERROR   1
6571
6572
6573  031036  020403                  64$:    CMP     R4,R3
6574  031040  001402                          BEQ     KK8
```

```
6575  031042  004737  031450                  JSR     PC,KKERR3
6576
6577  031046                          KK8:
6578  031046  012737  000014  177746          MOV     #MOM1,@#CONTRL
6579  031054  013703  031400                  MOV     KKPAT4,R3               ;SEE IF THE TEST PATTERN WAS
6580  031060  005037  177746                  CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
6581  031064  016204  000002                  MOV     2(R2),R4
6582
6583                                                                  ;SEE IF REFERENCE ADDRESS
6584  031070  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6585  031076  001011                          BNE     64$
6586                                                                  ;IF NOT ERROR!
6587  031100  010237  001636                  MOV     R2,$TMP2
6588  031104  062737  000002  001636          ADD     #2,$TMP2
6589  031112  012737  000001  001634          MOV     #1,$TMP1
6590  031120  104001                          ERROR   1
6591
6592
6593  031122  020403                  64$:    CMP     R4,R3
6594  031124  001402                          BEQ     KK10
6595  031126  004737  031464                  JSR     PC,KKERR4
6596
6597  031132  005737  031366          KK10:   TST     KKFLG1          ;SEE IF THIS PERMUTATION OF
6598  031136  001005                          BNE     KK11            ;THE TEST PATTERN HAS BEEN
6599  031140  012737  177777  031366          MOV     #-1,KKFLG1      ;WRITTEN BOTH UPWARD AND
6600  031146  000137  030170                  JMP     KK3             ;DOWNWARD. IF NOT, KKFLG IS 0,
6601                                                                  ;GO BACK TO WRITE IT DOWNWARD.
6602
6603  031152  005037  031366          KK11:   CLR     KKFLG1          ;GENERATE THE NEXT PERMUTATION
6604  031156  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;OF THE TEST PATTERN IN THE
6605                                                                  ;TEST TABLE
6606  031164  020527  031400                  CMP     R5,#KKPAT4
6607  031170  001011                          BNE     KK12
6608
6609  031172  011537  031402                  MOV     (R5),KKPAT5
6610  031176  013715  031374                  MOV     KKPAT2,(R5)
6611  031202  012705  031374                  MOV     #KKPAT2,R5
6612  031206  013715  031402                  MOV     KKPAT5,(R5)
6613  031212  000406                          BR      KK13
6614
6615  031214  012537  031402          KK12:   MOV     (R5)+,KKPAT5
6616  031220  011565  177776                  MOV     (R5),-2(R5)
6617  031224  013715  031402                  MOV     KKPAT5,(R5)
6618
6619  031230  005300                  KK13:   DEC     R0
6620  031232  001402                          BEQ     KK14
6621  031234  000137  030170                  JMP     KK3             ;GO DO NEXT PERMUTATION.
6622
6623  031240  012700  000006          KK14:   MOV     #6,R0
6624  031244  013737  031372  031402          MOV     KKPAT1,KKPAT5
6625  031252  005337  031370                  DEC     KKCNT1
6626
6627  031256  022737  000003  031370          CMP     #3,KKCNT1
6628  031264  001010                          BNE     KK15
6629
6630  031266  013737  031374  031372          MOV     KKPAT2,KKPAT1
```

```
6631  031274  013737  031402  031374          MOV     KKPAT5,KKPAT2
6632  031302  000137  030170                  JMP     KK3             ;GO DO NEXT PERMUTATION.
6633
6634  031306  022737  000002  031370  KK15:   CMP     #2,KKCNT1
6635  031314  001010                          BNE     KK16
6636
6637  031316  013737  031376  031372          MOV     KKPAT3,KKPAT1
6638  031324  013737  031402  031376          MOV     KKPAT5,KKPAT3
6639  031332  000137  030170                  JMP     KK3             ;GO DO NEXT PERMUTION.
6640
6641  031336  022737  000001  031370  KK16:   CMP     #1,KKCNT1
6642  031344  001073                          BNE     KK17            ;BRANCH IF DONE!
6643
6644  031346  013737  031400  031372          MOV     KKPAT4,KKPAT1
6645  031354  013737  031402  031400          MOV     KKPAT5,KKPAT4
6646  031362  000137  030170                  JMP     kK3             :GO DO NEXT PERMUTATION.
6647
6648
6649  031366  000000                  KKFLG1: .WORD   0               ;0 IF STORING PATTERN UPWARD
6650                                                                  ;1 IF STORING DOWNWARD.
6651
6652  031370  000000                  KKCNT1: .WORD   0               ;COUNTER USED IN GENERATING
6653                                                                  ;THE TEST PATTERN PERMUTATIONS.
6654
6655  031372  000000                  KKPAT1: .WORD   0               ;TEST PATTERN TABLE.
6656  031374  000000                  KKPAT2: .WORD   0
6657  031376  000000                  KKPAT3: .WORD   0
6658  031400  000000                  KKPAT4: .WORD   0
6659  031402  000000                  KKPAT5: .WORD   0
6660
6661  031404  000000                  KKTMP1: .WORD   0               ;USED TO LOCATE A TEST AREA WHOSE
6662  031406  000000  000000  000000  KKTMP2: .WORD   0,0,0,0         ;HITS WON'T BE WIPED OUT BY TEST CODE.
6663  031414  000000
6664
6665  031416  010137  001642          KKERR1: MOV     R1,$TMP4        ;ERROR REPORTING ROUTINES
6666  031422  005037  001640                  CLR     $TMP3
6667  031426  000427                          BR      KKERR5
6668
6669  031430  010137  001642          KKERR2: MOV     R1,$TMP4
6670  031434  062737  000002  001642          ADD     #2,$TMP4
6671  031442  005037  001640                  CLR     $TMP3
6672  031446  000417                          BR      KKERR5
6673
6674  031450  010237  001642          KKERR3: MOV     R2,$TMP4
6675  031454  013737  000001  001640          MOV     1,$TMP3
6676  031462  000411                          BR      KKERR5
6677
6678  031464  010237  001642          KKERR4: MOV     R2,$TMP4
6679  031470  062737  000002  001642          ADD     #2,$TMP4
6680  031476  012737  000001  001640          MOV     #1,$TMP3
6681  031504  000400                          BR      KKERR5
6682
6683  031506  010337  001636          KKERR5: MOV     R3,$TMP2
6684  031512  011637  001634                  MOV     (SP),$TMP1
6685  031516  012737  000014  177746          MOV     #MOM1,@#CONTRL
6686
```

```
                                ERROR   21

                                CLR     @#CONTRL
                                RTS     PC

                        KK17:   RSET                    ;DONE!

                        ;;*****************************************************************
                        ;*TEST 30       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.
                        ;*
                        ;;*****************************************************************
000004                  TST30:  SCOPE
012737  000040  001702          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                                        ;SET THE SKAD REGISTER
012737  033400  043632          MOV     #TST31,SKAD     ;IN CASE THE TEST ABORTS.

012737  043500  000114          MOV     #SPUR,@#CACHVEC ;ADDRESS AND PUT THE NO ERROR
113737  001502  001632          MOVB    $TSTNM,$TMP0    ;EXPECTED ROUTINES ADDRESS IN
                                                        ;THE PARITY ERROR VECTOR.

012737  001001  033236  MM1:     MOV     #001001,MMPAT1  ;SET UP THE PATTERN
012737  004004  033240          MOV     #004004,MMPAT2  ;REGISTERS.
012737  020020  033242          MOV     #020020,MMPAT3
012737  100100  033244          MOV     #100100,MMPAT4

012700  033250                  MOV     #MMTMP2,R0      ;LOCATE THE TEST AREA IN
042700  176003                  BIC     #176003,R0      ;MEMORY WHOSE 'HITS' WILL NOT
010001                          MOV     R0,R1           ;INTERFER WITH HITS CAUSED
062701  140000                  ADD     #TESTR1,R1      ;BY EXECUTING THIS TEST'S
010002                          MOV     R0,R2           ;CODE.
062702  142000                  ADD     #TESTR2,R2

010137  001644                  MOV     R1,$TMP5        ;SAVE THE TEST AREA ADDRESSES
010137  001646                  MOV     R1,$TMP6                ;FOR ERROR PRINT OUT.
062737  000002  001646          ADD     #2,$TMP6
010237  001650                  MOV     R2,$TMP7
010237  001652                  MOV     R2,$TMP10
062737  000002  001652          ADD     #2,$TMP10

012737  031706  001510          MOV     #MM2,$LPERR     ;SET THE LOOP ON ERROR REGISTER.

000240                  MM2:    NOP
012737  000034  177746          MOV     #SOMOM1,@#CONTRL        ;MAKE THE TEST AREAS HITS
005711                          TST     (R1)            ;IN GROUP 0 AND 1.
005761  000002                  TST     2(R1)
012737  000054  177746          MOV     #S1MOM1,@#CONTRL
```

```
6743  031732  005712                          TST     (R2)
6744  031734  005762  000002                  TST     2(R2)
6745  031740  005037  177746                  CLR     @#CONTRL
6746
6747
6748  031744  005711                          TST     (R1)
6749
6750                                                                ;SEE IF REFERENCE ADDRESS
6751  031746  032737  000010  177752          BIT     #10,@#HITMIS  ;IS A HIT.
6752  031754  001006                          BNE     MM3
6753                                                                ;IF NOT ERROR!
6754  031756  010137  001636                  MOV     R1,$TMP2
6755  031762  012737  000000  001634          MOV     #0,$TMP1
6756  031770  104001                          ERROR   1
6757
6758
6759
6760  031772                          MM3:
6761
6762  031772  005761  000002                  TST     2(R1)
6763
6764                                                                ;SEE IF REFERENCE ADDRESS
6765  031776  032737  000010  177752          BIT     #10,@#HITMIS  ;IS A HIT.
6766  032004  001011                          BNE     MM4
6767                                                                ;IF NOT ERROR!
6768  032006  010137  001636                  MOV     R1,$TMP2
6769  032012  062737  000002  001636          ADD     #2,$TMP2
6770  032020  012737  000000  001634          MOV     #0,$TMP1
6771  032026  104001                          ERROR   1
6772
6773
6774
6775  032030                          MM4:
6776
6777  032030  005712                          TST     (R2)
6778
6779                                                                ;SEE IF REFERENCE ADDRESS
6780  032032  032737  000010  177752          BIT     #10,@#HITMIS  ;IS A HIT.
6781  032040  001006                          BNE     MM5
6782                                                                ;IF NOT ERROR!
6783  032042  010237  001636                  MOV     R2,$TMP2
6784  032046  012737  000001  001634          MOV     #1,$TMP1
6785  032054  104001                          ERROR   1
6786
6787
6788
6789  032056                          MM5:
6790
6791  032056  005762  000002                  TST     2(R2)
6792
6793                                                                ;SEE IF REFERENCE ADDRESS
6794  032062  032737  000010  177752          BIT     #10,@#HITMIS  ;IS A HIT.
6795  032070  001014                          BNE     MM6
6796                                                                ;IF NOT ERROR!
6797  032072  010237  001636                  MOV     R2,$TMP2
6798  032076  062737  000002  001636          ADD     #2,$TMP2
```

```
 6799  032104  012737  000001  001634          MOV     #1,$TMP1
 6800  032112  104001                          ERROR   1
 6801
 6802
 6803
 6804  032114  012737  032122  001510          MOV     #MM6,$LPERR     ;SET LOOP ON ERROR ADDRESS
 6805  032122  012703  000001          MM6:    MOV     #1,R3
 6806  032126  012704  000004                  MOV     #4,R4
 6807  032132  110321                  MM7:    MOVB    R3,(R1)+        ;PUT THE TEST PATTERN
 6808  032134  006103                          ROL     R3              ;IN GROUP 0
 6809  032136  077403                          SOB     R4,MM7
 6810
 6811  032140  012704  000004                  MOV     #4,R4
 6812  032144  110322                  MM8:    MOVB    R3,(R2)+        ;PUT THE TEST PATTERN
 6813  032146  006103                          ROL     R3              ;IN GROUP 1
 6814  032150  077403                          SOB     R4,MM8
 6815  032152  010001                          MOV     R0,R1
 6816  032154  062701  140000                  ADD     #TESTR1,R1      ;RE-ESTABLISH POINTERS TO
 6817  032160  010002                          MOV     R0,R2           ;THE TEST LOCATIONS.
 6818  032162  062702  142000                  ADD     #TESTR2,R2
 6819  032166  012703  033236                  MOV     #MMPAT1,R3      ;PUT THE ADDRESS OF THE TEST
 6820                                                                  ;FATTERN REGISTERS IN R3
 6821
 6822  032172  005005                          CLR     R5
 6823
 6824
 6825  032174  005005                          CLR     R5
 6826  032176  111105                          MOVB    (R1),R5         ;GET THE PATTERN OUT OF
 6827  032200  032737  000010  177752          BIT     #10,@#HITMIS    ;THIS BYTE MAKING SURE IT
 6828  032206  001006                          BNE     MM9             ;IS A HIT
 6829  032210  010137  001636                  MOV     R1,$TMP2
 6830  032214  012737  000000  001634          MOV     #0,$TMP1
 6831  032222  104001                          ERROR   1
 6832
 6833  032224  042705  177400          MM9:    BIC     #177400,R5
 6834  032230  022705  000001                  CMP     #1,R5   ;SEE IF THE DATA IS CORRECT.
 6835  032234  001402                          BEQ     MM10
 6836  032236  004737  033260                  JSR     PC,MMERR1
 6837  032242  105121                  MM10:   COMB    (R1)+           ;COMPLIMENT THE TEST PATTERN
 6838  032244  012713  001376                  MOV     #001376,(R3)
 6839
 6840
 6841
 6842  032250  005005                          CLR     R5
 6843  032252  111105                          MOVB    (R1),R5         ;GET THE PATTERN OUT OF
 6844  032254  032737  000010  177752          BIT     #10,@#HITMIS    ;THIS BYTE MAKING SURE IT
 6845  032262  001006                          BNE     MM11            ;IS A HIT
 6846  032264  010137  001636                  MOV     R1,$TMP2
 6847  032270  012737  000000  001634          MOV     #0,$TMP1
 6848  032276  104001                          ERROR   1
 6849
 6850  032300  042705  177400          MM11:   BIC     #177400,R5
 6851  032304  022705  000002                  CMP     #2,R5   ;SEE IF THE DATA IS CORRECT.
 6852  032310  001402                          BEQ     MM12
 6853  032312  004737  033260                  JSR     PC,MMERR1
 6854  032316  105121                  MM12:   COMB    (R1)+           ;COMPLIMENT THE TEST PATTERN
```

```
6855  032320  012713  176776                    MOV     #176776,(R3)
6856
6857
6858  032324  062703  000002                    ADD     #2,R3           ;POINT TO THE NEXT ELEMENT
6859                                                                      ;IN THE TEST PATTERN TABLE.
6860
6861  032330  005005                            CLR     R5
6862  032332  111105                            MOVB    (R1),R5         ;GET THE PATTERN OUT OF
6863  032334  032737  000010  177752            BIT     #10,@#HITMIS    ;THIS BYTE MAKING SURE IT
6864  032342  001006                            BNE     MM13            ;IS A HIT
6865  032344  010137  001636                    MOV     R1,$TMP2
6866  032350  012737  000000  001634            MOV     #0,$TMP1
6867  032356  104001                            ERROR   1
6868
6869  032360  042705  177400          MM13:     BIC     #177400,R5
6870  032364  022705  000004                    CMP     #4,R5   ;SEE IF THE DATA IS CORRECT.
6871  032370  001402                            BEQ     MM14
6872  032372  004737  033260                    JSR     PC,MMERR1
6873  032376  105121                  MM14:     COMB    (R1)+           ;COMPLIMENT THE TEST PATTERN
6874  032400  012713  004373                    MOV     #004373,(R3)
6875
6876
6877
6878  032404  005005                            CLR     R5
6879  032406  111105                            MOVB    (R1),R5         ;GET THE PATTERN OUT OF
6880  032410  032737  000010  177752            BIT     #10,@#HITMIS    ;THIS BYTE MAKING SURE IT
6881  032416  001006                            BNE     MM15            ;IS A HIT
6882  032420  010137  001636                    MOV     R1,$TMP2
6883  032424  012737  000000  001634            MOV     #0,$TMP1
6884  032432  104001                            ERROR   1
6885
6886  032434  042705  177400          MM15:     BIC     #177400,R5
6887  032440  022705  000010                    CMP     #10,R5  ;SEE IF THE DATA IS CORRECT.
6888  032444  001402                            BEQ     MM16
6889  032446  004737  033260                    JSR     PC,MMERR1
6890  032452  105121                  MM16:     COMB    (R1)+           ;COMPLIMENT THE TEST PATTERN
6891  032454  012713  173773                    MOV     #173773,(R3)
6892
6893
6894  032460  062703  000002                    ADD     #2,R3           ;POINT TO THE NEXT ELEMENT
6895                                                                      ;IN THE TEST PATTERN TABLE.
6896
6897  032464  005005                            CLR     R5
6898  032466  111205                            MOVB    (R2),R5         ;GET THE PATTERN OUT OF
6899  032470  032737  000010  177752            BIT     #10,@#HITMIS    ;THIS BYTE MAKING SURE IT
6900  032476  001006                            BNE     MM17            ;IS A HIT
6901  032500  010237  001636                    MOV     R2,$TMP2
6902  032504  012737  000001  001634            MOV     #1,$TMP1
6903  032512  104001                            ERROR   1
6904
6905  032514  042705  177400          MM17:     BIC     #177400,R5
6906  032520  022705  000020                    CMP     #20,R5  ;SEE IF THE DATA IS CORRECT.
6907  032524  001402                            BEQ     MM18
6908  032526  004737  033272                    JSR     PC,MMERR2
6909  032532  105122                  MM18:     COMB    (R2)+           ;COMPLIMENT THE TEST PATTERN
6910  032534  012713  020357                    MOV     #020357,(R3)
```

```
6911
6912
6913
6914  032540  005005                          CLR     R5
6915  032542  111205                          MOVB    (R2),R5          ;GET THE PATTERN OUT OF
6916  032544  032737  000010  177752          BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
6917  032552  001006                          BNE     MM19             ;IS A HIT
6918  032554  010237  001636                  MOV     R2,$TMP2
6919  032560  012737  000001  001634          MOV     #1,$TMP1
6920  032566  104001                          ERROR   1
6921
6922  032570  042705  177400          MM19:   BIC     #177400,R5
6923  032574  022705  000040                  CMP     #40,R5  ;SEE IF THE DATA IS CORRECT.
6924  032600  001402                          BEQ     MM20
6925  032602  004737  033272                  JSR     PC,MMERR2
6926  032606  105122                  MM20:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
6927  032610  012713  157757                  MOV     #157757,(R3)
6928
6929
6930  032614  062703  000002                  ADD     #2,R3            ;POINT TO THE LAST ELEMENT
6931                                                                   ;IN THE TEST PATTERN TABLE.
6932
6933  032620  005005                          CLR     R5
6934  032622  111205                          MOVB    (R2),R5          ;GET THE PATTERN OUT OF
6935  032624  032737  000010  177752          BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
6936  032632  001006                          BNE     MM21             ;IS A HIT
6937  032634  010237  001636                  MOV     R2,$TMP2
6938  032640  012737  000001  00'   :         MOV     #1,$TMP1
6939  032646  104001                          ERROR   1
6940
6941  032650  042705  177400          MM21:   BIC     #177400,R5
6942  032654  022705  000100                  CMP     #100,R5 ;SEE IF THE DATA IS CORRECT.
6943  032660  001402                          BEQ     MM22
6944  032662  004737  033272                  JSR     PC,MMERR2
6945  032666  105122                  MM22:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
6946  032670  012713  100277                  MOV     #100277,(R3)
6947
6948
6949
6950  032674  005005                          CLR     R5
6951  032676  111205                          MOVB    (R2),R5          ;GET THE PATTERN OUT OF
6952  032700  032737  000010  177752          BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
6953  032706  001006                          BNE     MM23             ;IS A HIT
6954  032710  010237  001636                  MOV     R2,$TMP2
6955  032714  012737  000001  001634          MOV     #1,$TMP1
6956  032722  104001                          ERROR   1
6957
6958  032724  042705  177400          MM23:   BIC     #177400,R5
6959  032730  022705  000200                  CMP     #200,R5 ;SEE IF THE DATA IS CORRECT.
6960  032734  001402                          BEQ     MM24
6961  032736  004737  033272                  JSR     PC,MMERR2
6962  032742  105122                  MM24:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
6963  032744  012713  077677                  MOV     #077677,(R3)
6964
6965
6966  032750  010001                          MOV     R0,R1            ;RE-ESTABLISH POINTERS TO
```

```
6967  032752  062701  140000                  ADD     #TESTR1,R1      ;THE TEST AREA
6968  032756  010002                          MOV     R0,R2
6969  032760  062702  142000                  ADD     #TESTR2,R2
6970
6971
6972  032764  012105                          MOV     (R1)+,R5        ;CHECK THE COMPLIMENTED
6973
6974  032766  005761  177776                  TST     -2(R1)
6975
6976                                                                  ;SEE IF REFERENCE ADDRESS
6977  032772  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
6978  033000  001011                          BNE     MM25
6979                                                                  ;IF NOT ERROR!
6980  033002  010137  001636                  MOV     R1,$TMP2
6981  033006  062737  177776  001636          ADD     #-2,$TMP2
6982  033014  012737  000000  001634          MOV     #0,$TMP1
6983  033022  104001                          ERROR   1
6984
6985
6986
6987
6988  033024  020537  033236          MM25:   CMP     R5,MMPAT1               ;IS PATTERN CORRECT?
6989  033030  001402                          BEQ     MM26
6990  033032  004737  033322                  JSR     PC,MMERR4
6991
6992
6993  033036                          MM26:
6994
6995  033036  012105                          MOV     (R1)+,R5        ;CHECK THE COMPLIMENTED
6996
6997  033040  005761  177776                  TST     -2(R1)
6998
6999                                                                  ;SEE IF REFERENCE ADDRESS
7000  033044  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
7001  033052  001011                          BNE     MM27
7002                                                                  ;IF NOT ERROR!
7003  033054  010137  001636                  MOV     R1,$TMP2
7004  033060  062737  177776  001636          ADD     #-2,$TMP2
7005  033066  012737  000000  001634          MOV     #0,$TMP1
7006  033074  104001                          ERROR   1
7007
7008
7009
7010
7011  033076  020537  033240          MM27:   CMP     R5,MMPAT2               ;IS PATTERN CORRECT?
7012  033102  001402                          BEQ     MM28
7013  033104  004737  033322                  JSR     PC,MMERR4
7014
7015
7016  033110                          MM28:
7017
7018  033110  012205                          MOV     (R2)+,R5        ;CHECK THE COMPLIMENTED
7019
7020  033112  005762  177776                  TST     -2(R2)
7021
7022                                                                  ;SEE IF REFERENCE ADDRESS
```

```
7023  033116  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
7024  033124  001011                          BNE     MM29
7025                                                                  ;IF NOT ERROR!
7026  033126  010237  001636                  MOV     R2,$TMP2
7027  033132  062737  177776  001636          ADD     #-2,$TMP2
7028  033140  012737  000001  001634          MOV     #1,$TMP1
7029  033146  104001                          ERROR   1
7030
7031
7032
7033
7034  033150  020537  033242          MM29:   CMP     R5,MMPAT3               ;IS PATTERN CORRECT?
7035  033154  001402                          BEQ     MM30
7036  033156  004737  033342                  JSR     PC,MMERR5
7037
7038
7039  033162                          MM30:
7040
7041  033162  012205                          MOV     (R2)+,R5        ;CHECK THE COMPLIMENTED
7042
7043  033164  005762  177776                  TST     -2(R2)
7044
7045                                                                  ;SEE IF REFERENCE ADDRESS
7046  033170  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
7047  033176  001011                          BNE     MM31
7048                                                                  ;IF NOT ERROR!
7049  033200  010237  001636                  MOV     R2,$TMP2
7050  033204  062737  177776  001636          ADD     #-2,$TMP2
7051  033212  012737  000001  001634          MOV     #1,$TMP1
7052  033220  104001                          ERROR   1
7053
7054
7055
7056
7057  033222  020537  033244          MM31:   CMP     R5,MMPAT4               ;IS PATTERN CORRECT?
7058  033226  001464                          BEQ     MM32
7059  033230  004737  033342                  JSR     PC,MMERR5
7060
7061  033234  000461                          BR      MM32            ;FINISHED THIS TEST.
7062
7063  033236  000000          MMPAT1: .WORD   0               ;THIS IS THE TEST PATTERN
7064  033240  000000          MMPAT2: .WORD   0               ;TABLE.
7065  033242  000000          MMPAT3: .WORD   0
7066  033244  000000          MMPAT4: .WORD   0
7067
7068  033246  000000          MMTMP1: .WORD   0               ;THIS AREA IS USED TO ESTABLISH
7069  033250  000004          MMTMP2: .BLKW   4               ;A TEST LOCATION WHOSE HITS WON'T
7070                                                          ;BE INTERFERRED WITH BY THE CODE
7071                                                          ;IN THE REST OF THIS TEST.
7072
7073  033260  005037  001634          MMERR1: CLR     $TMP1           ;COME HERE TO REPORT
7074  033264  010137  001642                  MOV     R1,$TMP4        ;GROUP 0 ERROR,WHILE READING
7075  033270  000405                          BR      MMERR3          ;A BYTE INTO R5
7076
7077  033272  012737  000001  001634  MMERR2: MOV     #1,$TMP1                ;COME HERE TO REPORT
7078  033300  010237  001642                  MOV     R2,$TMP4        ;GROUP 1 ERROR, READING A
```

```
                                                                ;BYTE INTO R5.
033304  012637  001636          MMERR3: MOV     (SP)+,$TMP2
033310  010537  001640                  MOV     R5,$TMP3

033314  104017                          ERROR   17
033316  000177  146314                  JMP     @$TMP2

033322  005037  001634          MMERR4: CLR     $TMP1           ;REPORT AN ERROR IN GROUP
033326  010137  001642                  MOV     R1,$TMP4        ;0 WHILE READING A WORD
033332  062737  177776  001642          ADD     #-2,$TMP4
033340  000410                          BR      MMERR6

033342  012737  000001  001634  MMERR5: MOV     #1,$TMP1
033350  010237  001642                  MOV     R2,$TMP4
033354  062737  177776  001642          ADD     #-2,$TMP4

033362  012637  001636          MMERR6: MOV     (SP)+,$TMP2
033366  010537  001640                  MOV     R5,$TMP3

033372  104020                          ERROR   20
033374  000177  146236                  JMP     @$TMP2

033400                          MM32:                           ;DONE!


;*****************************************************************
;*TEST 31        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.
                                ;*
                                ;*
                                ;;*****************************************************************
033400  000004                  TST31:  SCOPE
                                                                ;SET THE SKAD REGISTER
033402  012737  040046  043632          MOV     #TST32,SKAD     ;IN CASE THE TEST ABORTS.

033410  104407                          RSET
033412  113737  001502  001632          MOVB    $TSTNM,$TMPO

033420  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
033424  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
033430  012702  172300                  MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
033434  012703  000010                  MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
033440  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
033442  077302                          SOB     R3,64$          ;ADDRESSES.
033444  005020                          CLR     (R0)+
033446  012720  000200                  MOV     #200,(R0)+
033452  012720  000400                  MOV     #400,(R0)+
033456  012720  000600                  MOV     #600,(R0)+
033462  012720  001000                  MOV     #1000,(R0)+
033466  012720  001200                  MOV     #1200,(R0)+
033472  012720  001400                  MOV     #1400,(R0)+
033476  012710  177600                  MOV     #177600,(R0)

033502  012737  000001  177572          MOV     #1,@#MMR0
033510  012737  000060  172516          MOV     #60,@#MMR3

033516  004737  037612          INTO:   JSR     PC,GTBINT               ;INITIALIZE THE MEMORY BUFFER
                                                                        ;ALLOCATION ROUTINES.
```

```
7191  033522  004737  045046                  JSR     PC,SIZDEV               ;GO DETERMINE WHAT DEVICES ARE
7192                                                                          ;PRESENT.
7193  033526  005046                          CLR     -(SP)                   ;MAKE THE WAIT LOOP ACCESSABLE
7194  033530  012746  034000                  MOV     #WAITLP,-(SP)           ;TO AN 'RTI'.
7195
7196  033534  012700  045440          INT1:   MOV     #RS4DFL,R0              ;GET READY TO SEE WHAT DEVICES
7197  033540  012701  033730                  MOV     #RS4CR,R1               ;ARE TO BE USED.
7198  033544  012702  033742                  MOV     #RS4SUM,R2
7199  033550  012703  033754                  MOV     #RS4ASS,R3
7200  033554  012704  000005                  MOV     #5,R4
7201
7202  033560  005011          INT2:   CLR     (R1)                    ;CLEAR THE UNIT NUMBER.
7203  033562  005012                  CLR     (R2)                    ;CLEAR THE COUNTER.
7204  033564  105710                  TSTB    (R0)                    ;IS THERE A DRIVE.
7205  033566  001447                  BEQ     INT6                    ;BRANCH IF NOT.
7206
7207  033570  111005                  MOVB    (R0),R5                 ;OTHERWISE DETERMINE A UNIT NUM.
7208  033572  104405                  SAVREG
7209  033574  012700  000010                  MOV     #10,R0
7210  033600  005001                  CLR     R1
7211  033602  012702  000001                  MOV     #1,R2
7212  033606  030205          INT3:   BIT     R2,R5
7213  033610  001405                  BEQ     INT4
7214  033612  010137  033724                  MOV     R1,INTMP1
7215  033616  104406                  RESREG
7216  033620  000137  033640                  JMP     INT5
7217  033624  005201          INT4:   INC     R1
7218  033626  006302                  ASL     R2
7219  033630  077012                  SOB     R0,INT3
7220  033632  104406                  RESREG
7221  033634  000137  033706                  JMP     INT6
7222
7223  033640  013711  033726  INT5:   MOV     CYCNT,(R1)              ;FOUND THE DRIVE SO SET UP THE
7224  033644  020127  033730                  CMP     R1,#RS4CR
7225  033650  001001                  BNE     1$
7226  033652  006311                  ASL     (R1)
7227  033654  020127  033732  1$:     CMP     R1,#RP4CR
7228  033660  001001                  BNE     2$
7229  033662  006311                  ASL     (R1)
7230  033664  020127  033734  2$:     CMP     R1,#RH4CR
7231  033670  001001                  BNE     3$
7232  033672  006311                  ASL     (R1)
7233  033674  012746  000340  3$:     MOV     #340,-(SP)              ;PASS COUNT AND MAKE THE DRIVER
7234  033700  011346                  MOV     (R3),-(SP)              ;ACCESSIBLE BY A 'RTI'.
7235  033702  013712  033724                  MOV     INTMP1,(R2)
7236
7237  033706  005200          INT6:   INC     R0
7238  033710  005721                  TST     (R1)+           ;MOVE THE POINTERS TO THE NEXT DEVICE.
7239  033712  022223                  CMP     (R2)+,(R3)+
7240  033714  000240                  NOP
7241  033716  077460                  SOB     R4,INT2
7242
7243
7244  033720  000240                  NOP
7245  033722  000002                  RTI                             ;START THE TEST!
7246
```

```
                                        ;THESE ARE SOME TABLES THAT ARE USED TO CONTROL AND SET UP THIS TEST.
033724  000000                  INTMPI: .WORD   0

033726  000010                  CYCNT:  .WORD   10              ;THE PASS COUNT!!!!

033730  000000                  RS4CR:  .WORD   0               ;PASS COUNT FOR EACH DEVICE.
033732  000000                  RP4CR:  .WORD   0
033734  000000                  RH4CR:  .WORD   0
033736  000000                  RK5CR:  .WORD   0
033740  000000                  UBECR:  .WORD   0

033742  000000                  RS4SUN: .WORD   0               ;THE DRIVE NUMBER USED FOR EACH
033744  000000                  RP4SUN: .WORD   0               ;DEVICE.
033746  000000                  RH4SUN: .WORD   0
033750  000000                  RK5SUN: .WORD   0
033752  000000                  UBESUN: .WORD   0

        033754                  SETBLE=RS4ASS
033754  034044                  RS4ASS: .WORD   DRRS4           ;STARTING ADDRESSES OF EACH DRIVER.
033756  034656                  RP4ASS: .WORD   DRRP4
033760  035470                  RH4ASS: .WORD   DRRH4
033762  036262                  RK5ASS: .WORD   DRRK5
033764  037074                  UBEASS: .WORD   DRUBE

033766  000000                  RS4RB:  .WORD   0               ;WRITE AND READ BUFFERS OF EACH DEVICE.
033770  000000                  RP4RB:  .WORD   0
033772  000000                  RH4RB:  .WORD   0
033774  000000                  RK5RB:  .WORD   0
033776  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.
034000  000230                  WAITLP: SPL     0               ;LOWER THE PRIORITY.
034002  005737  033736                  TST     RK5CR           ;WAIT FOR INTERRUPT OR ZERO PASS COUNT.
034006  001374                          BNE     WAITLP
034010  005737  033740                  TST     UBECR
034014  001371                          BNE     WAITLP
034016  005737  033732                  TST     RP4CR
034022  001366                          BNE     WAITLP
034024  005737  033730                  TST     RS4CR
034030  001363                          BNE     WAITLP
034032  005737  033734                  TST     RH4CR
034036  001360                          BNE     WAITLP

034040  000137  040044                  JMP     INDONE          ;FINISHED!!!


                                ;THIS IS THE RS4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                ;TEST.

034044  000240                  DRRS4:  NOP
034046  012737  007007  034652          MOV     #7007,DRS4T1    ;INITIALIZE THE RANDOM DISK ADDRESS
```

```
 7303  034054  012737  006006  034654          MOV     #6006,DRS4T2    ;GENERATER.
 7304  034062  012737  005005  034326          MOV     #5005,RS4AA3
 7305
 7306  034070  000240                  RS4AA:  NOP
 7307  034072  000240                          NOP
 7308  034074  104405                          SAVREG
 7309  034076  004737  037726                  JSR     PC,GETBUF       ;GET A MEMORY BUFFER.
 7310  034102  033766                          .WORD   RS4RB
 7311  034104  013701  033766                  MOV     RS4RB,R1
 7312  034110  005000                          CLR     R0
 7313  034112  073027  000014                  ASHC    #12.,R0
 7314
 7315  034116  000237                          SPL     7               ;GET A RANDOM DISK ADDRESS.
 7316  034120  013737  034652  043056          MOV     DRS4T1,$HINUM
 7317  034126  013737  034654  043060          MOV     DRS4T2,$LONUM
 7318  034134  004737  042760                  JSR     PC,$RAND
 7319  034140  013737  043056  034652          MOV     $HINUM,DRS4T1
 7320  034146  013737  043060  034654          MOV     $LONUM,DRS4T2
 7321  034154  000230                          SPL     0
 7322
 7323  034156  013702  033742                  MOV     RS4SUN,R2               ;SET UP THE DEVICE UNIT NUM.
 7324  034162  110237  034523                  MOVB    R2,RS4112
 7325  034166  110237  034351                  MOVB    R2,RS4BB
 7326  034172  110237  034415                  MOVB    R2,RS4HH
 7327  034176  110237  034461                  MOVB    R2,RS4NN
 7328
 7329  034202  013703  034652                  MOV     DRS4T1,R3               ;SET UP THE DISK ADDRESS.
 7330  034206  013704  034654                  MOV     DRS4T2,R4
 7331  034212  010337  034352                  MOV     R3,RS4CC
 7332  034216  010337  034524                  MOV     R3,RS4113
 7333  034222  010337  034416                  MOV     R3,RS4II
 7334  034226  010337  034462                  MOV     R3,RS400
 7335  034232  010437  034354                  MOV     R4,RS4DD
 7336  034236  010437  034420                  MOV     R4,RS4JJ
 7337  034242  010437  034526                  MOV     R4,RS4114
 7338  034246  010437  034464                  MOV     R4,RS4PP
 7339
 7340  034252  010137  034330                  MOV     R1,RS4AA1               ;SET THE MEMORY ADDRESS.
 7341  034256  010137  034356                  MOV     R1,RS4EE
 7342  034262  010137  034422                  MOV     R1,RS4KK
 7343  034266  010137  034466                  MOV     R1,RS4QQ
 7344  034272  010137  034530                  MOV     R1,RS4115
 7345  034276  010037  034470                  MOV     R0,RS4RR
 7346  034302  010037  034532                  MOV     R0,RS4116
 7347  034306  010037  034332                  MOV     R0,RS4AA2
 7348  034312  010037  034360                  MOV     R0,RS4FF
 7349  034316  010037  034424                  MOV     R0,RS4LL
 7350
 7351  034322  104406                          RESREG
 7352
 7353  034324  104420                          WRRAND                  ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
 7354  034326  000000          RS4AA3: .WORD   0
 7355  034330  000000          RS4AA1: .WORD   0
 7356  034332  000000          RS4AA2: .WORD   0
 7357  034334  004000                  .WORD   4000
 7358  034336  005237  034326          INC     RS4AA3
```

```
7359
7360  034342  000240                          NOP
7361  034344  000237                          SPL     7
7362  034346  104421                          CALRS4          ;GET THE RS4 TO DO THE TRANSFER FROM MEMORY
7363  034350     161                          .BYTE   161
7364  034351     000          RS4BB:          .BYTE   0
7365  034352  000000          RS4CC:          .WORD   0
7366  034354  000000          RS4DD:          .WORD   0
7367  034356  000000          RS4EE:          .WORD   0
7368  034360  000000          RS4FF:          .WORD   0
7369  034362  004000                          .WORD   4000
7370  034364  034402                          .WORD   RS4GG
7371
7372  034366  000240                          NOP
7373  034370  004737  034600                  JSR     PC,RS4YY
7374  034374  005066  000002                  CLR     2(SP)
7375  034400  000002                          RTI                     ;GO DO SOMETHING ELSE WHILE WAITING
7376                                                                  ;FOR THE INTERRUPT!
7377
7378  034402  000240          RS4GG:          NOP
7379  034404  004737  034600                  JSR     PC,RS4YY                ;SEE IF THERE WERE ANY ERRORS.
7380
7381  034410  000237                          SPL     7
7382  034412  104421                          CALRS4          ;DO THE WRITE CHECK
7383  034414     151                          .BYTE   151
7384  034415     000          RS4HH:          .BYTE   0
7385  034416  000000          RS4II:          .WORD   0
7386  034420  000000          RS4JJ:          .WORD   0
7387  034422  000000          RS4KK:          .WORD   0
7388  034424  000000          RS4LL:          .WORD   0
7389  034426  004000                          .WORD   4000
7390  034430  034446                          .WORD   RS4MM
7391
7392  034432  000240                          NOP
7393  034434  004737  034600                  JSR     PC,RS4YY
7394  034440  005066  000002                  CLR     2(SP)
7395  034444  000002                          RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.
7396
7397  034446  000240          RS4MM:          NOP
7398  034450  004737  034600                  JSR     PC,RS4YY        ;SEE IF THERE WERE ANY ERRORS.
7399
7400
7401  034454  000237                          SPL     7
7402  034456  104421                          CALRS4          ;READ THE DISK.
7403  034460     171                          .BYTE   171
7404  034461     000          RS4NN:          .BYTE   0
7405  034462  000000          RS400:          .WORD   0
7406  034464  000000          RS4PP:          .WORD   0
7407  034466  000000          RS4QQ:          .WORD   0
7408  034470  000000          RS4RR:          .WORD   0
7409  034472  004000                          .WORD   4000
7410  034474  034512                          .WORD   RS4111
7411
7412  034476  000240                          NOP
7413  034500  004737  034600                  JSR     PC,RS4YY
7414  034504  005066  000002                  CLR     2(SP)
```

```
7415  034510  000002                          RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.
7416
7417  034512  004737  034600        RS4111: JSR     PC,RS4YY
7418  034516  000237                SPL     7
7419
7420  034520  104421                CALRS4
7421  034522     151                .BYTE   151
7422  034523     000        RS4112: .BYTE   0
7423  034524  000000        RS4113: .WORD   0
7424  034526  000000        RS4114: .WORD   0
7425  034530  000000        RS4115: .WORD   0
7426  034532  000000        RS4116: .WORD   0
7427  034534  004000                .WORD   4000
7428  034536  034554                .WORD   RS4SS
7429  034540  000240                NOP
7430  034542  004737  034600        JSR     PC,RS4YY
7431  034546  005066  000002        CLR     2(SP)
7432  034552  000002                RTI
7433
7434  034554  000240        RS4SS.  NOP
7435  034556  004737  034600        JSR     PC,RS4YY        ;SEE IF ANY ERRORS OCCURRED.
7436
7437  034562  005337  033730        DEC     RS4CR           ;DECRIMENT THE PASS COUNT.
7438  034566  001001                BNE     RS4XX           ;IF NOT DONE CONTINUE.
7439  034570  000002                RTI                     ;IF DONE GET OUT!
7440
7441  034572  000240        RS4XX:  NOP
7442  034574  000137  034070        JMP     RS4AA           ;RESTART.
7443
7444  034600  000240        RS4YY:  NOP
7445  034602  005737  046756        TST     RS4ER1  ;SEE IF ANY ERRORS OCCURRED.
7446  034606  001420                BEQ     RS4ZZ           ;IF NOT THEN RETURN TO CALL.
7447
7448  034610  000237                SPL     7
7449  034612  005037  033730        CLR     RS4CR           ;IF YES THEN CLEAR THE PASS CCUNT.
7450  034616  013737  046760  001634  MOV     RS4ER2,$TMP1    ;AND MAKE AN ERROR CALL.
7451  034624  013737  046764  001640  MOV     RS4ER4,$TMP3
7452  034632  013737  046762  001636  MOV     RS4ER3,$TMP2
7453  034640  104154                ERROR   154
7454  034642  000230                SPL     0
7455  034644  005726                TST     (SP)+
7456  034646  000002                RTI                     ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
7457                                                        ;FROM THE TEST.
7458
7459  034650  000207        RS4ZZ:  RTS     PC              ;THERE WERE NO ERRORS.
7460
7461  034652  000000        DRS4T1: .WORD   0
7462  034654  000000        DRS4T2: .WORD   0
7463
7464
7465
7466                        ;THIS IS THE RP4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
7467                        ;TEST.
7468
7469  034656  000240        DRRP4:  NOP
7470  034660  012737  004004  035464        MOV     #4004,DRP4T1    ;INITIALIZE THE RANDOM DISK ADDRESS
```

```
7471  034666  012737  003003  035466          MOV     #3003,DRP4T2    ;GENERATER.
7472  034674  012737  002002  035140          MOV     #2002,RP4AA3
7473
7474  034702  000240                  RP4AA:  NOP
7475  034704  000240                          NOP
7476  034706  104405                          SAVREG
7477  034710  004737  037726                  JSR     PC,GETBUF       ;GET A MEMORY BUFFER.
7478  034714  033770                          .WORD   RP4RB
7479  034716  013701  033770                  MOV     RP4RB,R1
7480  034722  005000                          CLR     R0
7481  034724  073027  000014                  ASHC    #12.,R0
7482
7483  034730  000237                          SPL     7               ;GET A RANDOM DISK ADDRESS.
7484  034732  013737  035464  043056          MOV     DRP4T1,$HINUM
7485  034740  013737  035466  043060          MOV     DRP4T2,$LONUM
7486  034746  004737  042760                  JSR     PC,$RAND
7487  034752  013737  043056  035464          MOV     $HINUM,DRP4T1
7488  034760  013737  043060  035466          MOV     $LONUM,DRP4T2
7489  034766  000230                          SPL     0
7490
7491  034770  013702  033744                  MOV     RP4SUN,R2               ;SET UP THE DEVICE UNIT NUM.
7492  034774  110237  035335                  MOVB    R2,RP4112
7493  035000  110237  035163                  MOVB    R2,RP4BB
7494  035004  110237  035227                  MOVB    R2,RP4HH
7495  035010  110237  035273                  MOVB    R2,RP4NN
7496
7497  035014  013703  035464                  MOV     DRP4T1,R3               ;SET UP THE DISK ADDRESS.
7498  035020  013704  035466                  MOV     DRP4T2,R4
7499  035024  010337  035164                  MOV     R3,RP4CC
7500  035030  010337  035336                  MOV     R3,RP4113
7501  035034  010337  035230                  MOV     R3,RP4II
7502  035040  010337  035274                  MOV     R3,RP4OO
7503  035044  010437  035166                  MOV     R4,RP4DD
7504  035050  010437  035232                  MOV     R4,RP4JJ
7505  035054  010437  035340                  MOV     R4,RP4114
7506  035060  010437  035276                  MOV     R4,RP4PP
7507
7508  035064  010137  035142                  MOV     R1,RP4AA1               ;SET THE MEMORY ADDRESS.
7509  035070  010137  035170                  MOV     R1,RP4EE
7510  035074  010137  035234                  MOV     R1,RP4KK
7511  035100  010137  035300                  MOV     R1,RP4QQ
7512  035104  010137  035342                  MOV     R1,RP4115
7513  035110  010037  035302                  MOV     R0,RP4RR
7514  035114  010037  035344                  MOV     R0,RP4116
7515  035120  010037  035144                  MOV     R0,RP4AA2
7516  035124  010037  035172                  MOV     R0,RP4FF
7517  035130  010037  035236                  MOV     R0,RP4LL
7518
7519  035134  104406                          RESREG
7520
7521  035136  104420                          WRRAND                  ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
7522  035140  000000          RP4AA3: .WORD   0
7523  035142  000000          RP4AA1: .WORD   0
7524  035144  000000          RP4AA2: .WORD   0
7525  035146  004000                  .WORD   4000
7526  035150  005237  035140          INC     RP4AA3
```

```
7527
7528  035154  000240                          NOP
7529  035156  000237                          SPL     7
7530  035160  104422                          CALRP4          ;GET THE RP4 TO DO THE TRANSFER FROM MEMORY
7531  035162     161                          .BYTE   161
7532  035163     000          RP4BB:          .BYTE   0
7533  035164  000000          RP4CC:          .WORD   0
7534  035166  000000          RP4DD:          .WORD   0
7535  035170  000000          RP4EE:          .WORD   0
7536  035172  000000          RP4FF:          .WORD   0
7537  035174  004000                          .WORD   4000
7538  035176  035214                          .WORD   RP4GG
7539
7540  035200  000240                          NOP
7541  035202  004737  035412                  JSR     PC,RP4YY
7542  035206  005066  000002                  CLR     2(SP)
7543  035212  000002                          RTI                     ;GO DO SOMETHING ELSE WHILE WAITING
7544                                                                  ;FOR THE INTERRUPT!
7545
7546  035214  000240          RP4GG:          NOP
7547  035216  004737  035412                  JSR     PC,RP4YY                . ;SEE IF THERE WERE ANY ERRORS.
7548
7549  035222  000237                          SPL     7
7550  035224  104422                          CALRP4          ;DO THE WRITE CHECK
7551  035226     151                          .BYTE   151
7552  035227     000          RP4HH:          .BYTE   0
7553  035230  000000          RP4II:          .WORD   0
7554  035232  000000          RP4JJ:          .WORD   0
7555  035234  000000          RP4KK:          .WORD   0
7556  035236  000000          RP4LL:          .WORD   0
7557  035240  004000                          .WORD   4000
7558  035242  035260                          .WORD   RP4MM
7559
7560  035244  000240                          NOP
7561  035246  004737  035412                  JSR     PC,RP4YY
7562  035252  005066  000002                  CLR     2(SP)
7563  035256  000002                          RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.
7564
7565  035260  000240          RP4MM:          NOP
7566  035262  004737  035412                  JSR     PC,RP4YY        ;SEE IF THERE WERE ANY ERRORS.
7567
7568
7569  035266  000237                          SPL     7
7570  035270  104422                          CALRP4          ;READ THE DISK.
7571  035272     171                          .BYTE   171
7572  035273     000          RP4NN:          .BYTE   0
7573  035274  000000          RP4OO:          .WORD   0
7574  035276  000000          RP4PP:          .WORD   0
7575  035300  000000          RP4QQ:          .WORD   0
7576  035302  000000          RP4RR:          .WORD   0
7577  035304  004000                          .WORD   4000
7578  035306  035324                          .WORD   RP4111
7579
7580  035310  000240                          NOP
7581  035312  004737  035412                  JSR     PC,RP4YY
7582  035316  005066  000002                  CLR     2(SP)
```

```
7583  035322  000002                          RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.
7584
7585  035324  004737  035412        RP4111: JSR     PC,RP4YY
7586  035330  000237                        SPL     7
7587
7588  035332  104422                        CALRP4
7589  035334    151                         .BYTE   151
7590  035335    000       RP4112:           .BYTE   0
7591  035336  000000      RP4113:           .WORD   0
7592  035340  000000      RP4114:           .WORD   0
7593  035342  000000      RP4115:           .WORD   0
7594  035344  000000      RP4116:           .WORD   0
7595  035346  004000                        .WORD   4000
7596  035350  035366                        .WORD   RP4SS
7597  035352  000240                        NOP
7598  035354  004737  035412                JSR     PC,RP4YY
7599  035360  005066  000002                CLR     2(SP)
7600  035364  000002                        RTI
7601
7602  035366  000240        RP4SS:          NOP
7603  035370  004737  035412                JSR     PC,RP4YY        ;SEE IF ANY ERRORS OCCURRED.
7604
7605  035374  005337  033732                DEC     RP4CR           ;DECRIMENT THE PASS COUNT.
7606  035400  001001                        BNE     RP4XX           ;IF NOT DONE CONTINUE.
7607  035402  000002                        RTI                     ;IF DONE GET OUT!
7608
7609  035404  000240        RP4XX:          NOP
7610  035406  000137  034702                JMP     RP4AA           ;RESTART.
7611
7612  035412  000240        RP4YY:          NOP
7613  035414  005737  046014                TST     RP4ER1  ;SEE IF ANY ERRORS OCCURRED.
7614  035420  001420                        BEQ     RP4ZZ           ;IF NOT THEN RETURN TO CALL.
7615
7616  035422  000237                        SPL     7
7617  035424  005037  033732                CLR     RP4CR           ;IF YES THEN CLEAR THE PASS COUNT.
7618  035430  013737  046016  001634        MOV     RP4ER2,$TMP1    ;AND MAKE AN ERROR CALL.
7619  035436  013737  046022  001640        MOV     RP4ER4,$TMP3
7620  035444  013737  046020  001636        MOV     RP4ER3,$TMP2
7621  035452  104155                        ERROR   155
7622  035454  000230                        SPL     0
7623  035456  005726                        TST     (SP)+
7624  035460  000002                        RTI                     ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
7625                                                                ;FROM THE TEST.
7626
7627  035462  000207        RP4ZZ:  RTS     PC              ;THERE WERE NO ERRORS.
7628
7629  035464  000000        DRP4T1: .WORD   0
7630  035466  000000        DRP4T2: .WORD   0
7631
7632
7633
7634                          ;THIS IS THE RH4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
7635                          ;TEST.
7636
7637  035470  000240        DRRH4:  NOP
7638  035472  012737  070070  036256        MOV     #70070,DRH4T1   ;INITIALIZE THE RANDOM DISK ADDRESS
```

```
 7639  035500  012737  060060  036260          MOV     #60060,DRH4T2   ;GENERATER.
 7640  035506  012737  050050  035732          MOV     #50050,RH4AA3
 7641
 7642  035514  000240                  RH4AA:  NOP
 7643  035516  000240                          NOP
 7644  035520  104405                          SAVREG
 7645  035522  004737  037726                  JSR     PC,GETBUF       ;GET A MEMORY BUFFER.
 7646  035526  033772                          .WORD   RH4RB
 7647  035530  013701  033772                  MOV     RH4RB,R1
 7648  035534  005000                          CLR     R0
 7649  035536  073027  000014                  ASHC    #12.,R0
 7650
 7651  035542  000237                          SPL     7               ;GET A RANDOM DISK ADDRESS.
 7652  035544  013737  036256  043056          MOV     DRH4T1,$HINUM
 7653  035552  013737  036260  043060          MOV     DRH4T2,$LONUM
 7654  035560  004737  042760                  JSR     PC,$RAND
 7655  035564  013737  043056  036256          MOV     $HINUM,DRH4T1
 7656  035572  013737  043060  036260          MOV     $LONUM,DRH4T2
 7657  035600  000230                          SPL     0
 7658
 7659  035602  013702  033746                  MOV     RH4SUN,R2               ;SET UP THE DEVICE UNIT NUM.
 7660  035606  110237  036127                  MOVB    R2,RH4112
 7661  035612  110237  035755                  MOVB    R2,RH4BB
 7662  035616  110237  036021                  MOVB    R2,RH4HH
 7663  035622  110237  036065                  MOVB    R2,RH4NN
 7664
 7665  035626  013703  036256                  MOV     DRH4T1,R3               ;SET UP THE DISK ADDRESS.
 7666  035632  013704  036260                  MOV     DRH4T2,R4
 7667  035636  010337  035756                  MOV     R3,RH4CC
 7668  035642  010337  036130                  MOV     R3,RH4113
 7669  035646  010337  036022                  MOV     R3,RH4II
 7670  035652  010337  036066                  MOV     R3,RH4OO
 7671
 7672  035656  010137  035734                  MOV     R1,RH4AA1               ;SET THE MEMORY ADDRESS.
 7673  035662  010137  035762                  MOV     R1,RH4EE
 7674  035666  010137  036026                  MOV     R1,RH4KK
 7675  035672  010137  036072                  MOV     R1,RH4QQ
 7676  035676  010137  036134                  MOV     R1,RH4115
 7677  035702  010037  036074                  MOV     R0,RH4RR
 7678  035706  010037  036136                  MOV     R0,RH4116
 7679  035712  010037  035736                  MOV     R0,RH4AA2
 7680  035716  010037  035764                  MOV     R0,RH4FF
 7681  035722  010037  036030                  MOV     R0,RH4LL
 7682
 7683  035726  104406                          RESREG
 7684
 7685  035730  104420                          WRRAND                  ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
 7686  035732  000000          RH4AA3: .WORD   0
 7687  035734  000000          RH4AA1: .WORD   0
 7688  035736  000000          RH4AA2: .WORD   0
 7689  035740  004000                  .WORD   4000
 7690  035742  005237  035732          INC     RH4AA3
 7691
 7692  035746  000240                  NOP
 7693  035750  000237                  SPL     7
 7694  035752  104423                  CALRH4          ;GET THE RH4 TO DO THE TRANSFER FROM MEMORY
```

```
035754     161                          .BYTE   161
035755     000                  RH4BB:  .BYTE   0
035756  000000                  RH4CC:  .WORD   0
035760  000000                  RH4DD:  .WORD   0
035762  000000                  RH4EE:  .WORD   0
035764  000000                  RH4FF:  .WORD   0
035766  004000                          .WORD   4000
035770  036006                          .WORD   RH4GG

035772  000240                          NOP
035774  004737  036204                  JSR     PC,RH4YY
036000  005066  000002                  CLR     2(SP)
036004  000002                          RTI                             ;GO DO SOMETHING ELSE WHILE WAITING
                                                                        ;FOR THE INTERRUPT!

036006  000240                  RH4GG:  NOP
036010  004737  036204                  JSR     PC,RH4YY                ;SEE IF THERE WERE ANY ERRORS.

036014  000237                          SPL     7
036016  104423                          CALRH4          ;DO THE WRITE CHECK
036020     171                          .BYTE   171
036021     000                  RH4HH:  .BYTE   0
036022  000000                  RH4II:  .WORD   0
036024  000000                  RH4JJ:  .WORD   0
036026  000000                  RH4KK:  .WORD   0
036030  000000                  RH4LL:  .WORD   0
036032  004000                          .WORD   4000
036034  036052                          .WORD   RH4MM

036036  000240                          NOP
036040  004737  036204                  JSR     PC,RH4YY
036044  005066  000002                  CLR     2(SP)
036050  000002                          RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.

036052  000240                  RH4MM:  NOP
036054  004737  036204                  JSR     PC,RH4YY        ;SEE IF THERE WERE ANY ERRORS.


036060  000237                          SPL     7
036062  104423                          CALRH4          ;READ THE DISK.
036064     151                          .BYTE   151
036065     000                  RH4NN:  .BYTE   0
036066  000000                  RH4OO:  .WORD   0
036070  000000                  RH4PP:  .WORD   0
036072  000000                  RH4QQ:  .WORD   0
036074  000000                  RH4RR:  .WORD   0
036076  004000                          .WORD   4000
036100  036116                          .WORD   RH4111

036102  000240                          NOP
036104  004737  036204                  JSR     PC,RH4YY
036110  005066  000002                  CLR     2(SP)
036114  000002                          RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.

036116  004737  036204          RH4111: JSR     PC,RH4YY
036122  000237                          SPL     7
```

```
 7751
 7752  036124  104423                    CALRH4
 7753  036126     171                    .BYTE   171
 7754  036127     000           RH4112:  .BYTE   0
 7755  036130  000000           RH4113:  .WORD   0
 7756  036132  000000           RH4114:  .WORD   0
 7757  036134  000000           RH4115:  .WORD   0
 7758  036136  000000           RH4116:  .WORD   0
 7759  036140  004000                    .WORD   4000
 7760  036142  036160                    .WORD   RH4SS
 7761  036144  000240                    NOP
 7762  036146  004737  036204            JSR     PC,RH4YY
 7763  036152  005066  000002            CLR     2(SP)
 7764  036156  000002                    RTI
 7765
 7766  036160  000240           RH4SS:   NOP
 7767  036162  004737  036204            JSR     PC,RH4YY        ;SEE IF ANY ERRORS OCCURRED.
 7768
 7769  036166  005337  033734            DEC     RH4CR           ;DECRIMENT THE PASS COUNT.
 7770  036172  001001                    BNE     RH4XX           ;IF NOT DONE CONTINUE.
 7771  036174  000002                    RTI                     ;IF DONE GET OUT!
 7772
 7773  036176  000240           RH4XX:   NOP
 7774  036200  000137  035514            JMP     RH4AA           ;RESTART.
 7775
 7776  036204  000240           RH4YY:   NOP
 7777  036206  005737  051460            TST     RH4ER1  ;SEE IF ANY ERRORS OCCURRED.
 7778  036212  001420                    BEQ     RH4ZZ           ;IF NOT THEN RETURN TO CALL.
 7779
 7780  036214  000237                    SPL     7
 7781  036216  005037  033734            CLR     RH4CR           ;IF YES THEN CLEAR THE PASS COUNT.
 7782  036222  013737  051462  001634    MOV     RH4ER2,$TMP1    ;AND MAKE AN ERROR CALL.
 7783  036230  013737  051466  001640    MOV     RH4ER4,$TMP3
 7784  036236  013737  051464  001636    MOV     RH4ER3,$TMP2
 7785  036244  104156                    ERRCR   156
 7786  036246  000230                    SPL     0
 7787  036250  005726                    TST     (SP)+
 7788  036252  000002                    RTI                     ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
 7789                                                            ;FROM THE TEST.
 7790
 7791  036254  000207           RH4ZZ:   RTS     PC              ;THERE WERE NO ERRORS.
 7792
 7793  036256  000000           DRH4T1:  .WORD   0
 7794  036260  000000           DRH4T2:  .WORD   0
 7795
 7796
 7797
 7798
 7799                           ;THIS IS THE RK5 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
 7800                           ;TEST.
 7801
 7802  036262  000240           DRRK5:   NOP
 7803  036264  012737  030030  037070    MOV     #30030,DRK5T1   ;INITIALIZE THE RANDOM DISK ADDRESS
 7804  036272  012737  040040  037072    MOV     #40040,DRK5T2   ;GENERATER.
 7805  036300  012737  050050  036544    MOV     #50050,RK5AA3
 7806
```

```
 7807  036306  000240                    RK5AA:  NOP
 7808  036310  000240                            NOP
 7809  036312  104405                            SAVREG
 7810  036314  004737  037726                    JSR     PC,GETBUF       ;GET A MEMORY BUFFER.
 7811  036320  033774                            .WORD   RK5RB
 7812  036322  013701  033774                    MOV     RK5RB,R1
 7813  036326  005000                            CLR     R0
 7814  036330  072027  000014                    ASHC    #12.,R0
 7815
 7816  036334  000237                            SPL     7               ;GET A RANDOM DISK ADDRESS.
 7817  036336  013737  037070  043056            MOV     DRK5T1,$HINUM
 7818  036344  013737  037072  043060            MOV     DRK5T2,$LONUM
 7819  036352  004737  042760                    JSR     PC,$RAND
 7820  036356  013737  043056  037070            MOV     $HINUM,DRK5T1
 7821  036364  013737  043060  037072            MOV     $LONUM,DRK5T2
 7822  036372  000230                            SPL     0
 7823
 7824  036374  013702  033750                    MOV     RK5SUN,R2               ;SET UP THE DEVICE UNIT NUM.
 7825  036400  110237  036741                    MOVB    R2,RK5112
 7826  036404  110237  036567                    MOVB    R2,RK5BB
 7827  036410  110237  036633                    MOVB    R2,RK5HH
 7828  036414  110237  036677                    MOVB    R2,RK5NN
 7829
 7830  036420  013703  037070                    MOV     DRK5T1,R3               ;SET UP THE DISK ADDRESS.
 7831  036424  013704  037072                    MOV     DRK5T2,R4
 7832  036430  010337  036570                    MOV     R3,RK5CC
 7833  036434  010337  036742                    MOV     R3,RK5113
 7834  036440  010337  036634                    MOV     R3,RK5II
 7835  036444  010337  036700                    MOV     R3,RK500
 7836  036450  010437  036572                    MOV     R4,RK5DD
 7837  036454  010437  036636                    MOV     R4,RK5JJ
 7838  036460  010437  036744                    MOV     R4,RK5114
 7839  036464  010437  036702                    MOV     R4,RK5PP
 7840
 7841  036470  010137  036546                    MOV     R1,RK5AA1               ;SET THE MEMORY ADDRESS.
 7842  036474  010137  036574                    MOV     R1,RK5EE
 7843  036500  010137  036640                    MOV     R1,RK5KK
 7844  036504  010137  036704                    MOV     R1,RK5QQ
 7845  036510  010137  036746                    MOV     R1,RK5115
 7846  036514  010037  036706                    MOV     R0,RK5RR
 7847  036520  010037  036750                    MOV     R0,RK5116
 7848  036524  010037  036550                    MOV     R0,RK5AA2
 7849  036530  010037  036576                    MOV     R0,RK5FF
 7850  036534  010037  036642                    MOV     R0,RK5LL
 7851
 7852  036540  104406                            RESREG
 7853
 7854  036542  104420                            WRRAND                  ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
 7855  036544  000000                RK5AA3: .WORD   0
 7856  036546  000000                RK5AA1: .WORD   0
 7857  036550  000000                RK5AA2: .WORD   0
 7858  036552  004000                        .WORD   4000
 7859  036554  005237  036544                INC     RK5AA3
 7860
 7861  036560  000240                        NOP
 7862  036562  000237                        SPL     7
```

```
 7863  036564  104424                          CALRK5          ;GET THE RK5 TO DO THE TRANSFER FROM MEMORY
 7864  036566     103                          .BYTE   103
 7865  036567     000                RK5BB:    .BYTE   0
 7866  036570  000000                RK5CC:    .WORD   0
 7867  036572  000000                RK5DD:    .WORD   0
 7868  036574  000000                RK5EE:    .WORD   0
 7869  036576  000000                RK5FF:    .WORD   0
 7870  036600  004000                          .WORD   4000
 7871  036602  036620                          .WORD   RK5GG
 7872
 7873  036604  000240                          NOP
 7874  036606  004737  037016                  JSR     PC,RK5YY
 7875  036612  005066  000002                  CLR     2(SP)
 7876  036616  000002                          RTI                     ;GO DO SOMETHING ELSE WHILE WAITING
 7877                                                                  ;FOR THE INTERRUPT!
 7878
 7879  036620  000240                RK5GG:    NOP
 7880  036622  004737  037016                  JSR     PC,RK5YY                ;SEE IF THERE WERE ANY ERRORS.
 7881
 7882  036626  000237                          SPL     7
 7883  036630  104424                          CALRK5          ;DO THE WRITE CHECK
 7884  036632     107                          .BYTE   107
 7885  036633     000                RK5HH:    .BYTE   0
 7886  036634  000000                RK5II:    .WORD   0
 7887  036636  000000                RK5JJ:    .WORD   0
 7888  036640  000000                RK5KK:    .WORD   0
 7889  036642  000000                RK5LL:    .WORD   0
 7890  036644  004000                          .WORD   4000
 7891  036646  036664                          .WORD   RK5MM
 7892
 7893  036650  000240                          NOP
 7894  036652  004737  037016                  JSR     PC,RK5YY
 7895  036656  005066  000002                  CLR     2(SP)
 7896  036662  000002                          RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.
 7897
 7898  036664  000240                RK5MM:    NOP
 7899  036666  004737  037016                  JSR     PC,RK5YY        ;SEE IF THERE WERE ANY ERRORS.
 7900
 7901
 7902  036672  000237                          SPL     7
 7903  036674  104424                          CALRK5          ;READ THE DISK.
 7904  036676     105                          .BYTE   105
 7905  036677     000                RK5NN:    .BYTE   0
 7906  036700  000000                RK5OO:    .WORD   0
 7907  036702  000000                RK5PP:    .WORD   0
 7908  036704  000000                RK5QQ:    .WORD   0
 7909  036706  000000                RK5RR:    .WORD   0
 7910  036710  004000                          .WORD   4000
 7911  036712  036730                          .WORD   RK5111
 7912
 7913  036714  000240                          NOP
 7914  036716  004737  037016                  JSR     PC,RK5YY
 7915  036722  005066  000002                  CLR     2(SP)
 7916  036726  000002                          RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.
 7917
 7918  036730  004737  037016        RK5111: JSR     PC,RK5YY
```

```
 7919  036734  000237                          SPL     7
 7920
 7921  036736  104424                          CALRK5
 7922  036740     107                          .BYTE   107
 7923  036741     000                RK5112:   .BYTE   0
 7924  036742  000000                RK5113:   .WORD   0
 7925  036744  000000                RK5114:   .WORD   0
 7926  036746  000000                RK5115:   .WORD   0
 7927  036750  000000                RK5116:   .WORD   0
 7928  036752  004000                          .WORD   4000
 7929  036754  036772                          .WORD   RK5SS
 7930  036756  000240                          NOP
 7931  036760  004737  037016                  JSR     PC,RK5YY
 7932  036764  005066  000002                  CLR     2(SP)
 7933  036770  000002                          RTI
 7934
 7935  036772  000240                RK5SS:    NOP
 7936  036774  004737  037016                  JSR     PC,RK5YY        ;SEE IF ANY ERRORS OCCURRED.
 7937
 7938  037000  005337  033736                  DEC     RK5CR           ;DECRIMENT THE PASS COUNT.
 7939  037004  001001                          BNE     RK5XX           ;IF NOT DONE CONTINUE.
 7940  037006  000002                          RTI                     ;IF DONE GET OUT!
 7941
 7942  037010  000240                RK5XX:    NOP
 7943  037012  000137  036306                  JMP     RK5AA           ;RESTART.
 7944
 7945  037016  000240                RK5YY:    NOP
 7946  037020  005737  047712                  TST     RK5ER1  ;SEE IF ANY ERRORS OCCURRED.
 7947  037024  001420                          BEQ     RK5ZZ           ;IF NOT THEN RETURN TO CALL.
 7948
 7949  037026  000237                          SPL     7
 7950  037030  005037  033736                  CLR     RK5CR           ;IF YES THEN CLEAR THE PASS COUNT.
 7951  037034  013737  047714  001634          MOV     RK5ER2,$TMP1    ;AND MAKE AN ERROR CALL.
 7952  037042  013737  047720  001640          MOV     RK5ER4,$TMP3
 7953  037050  013737  047716  001636          MOV     RK5ER3,$TMP2
 7954  037056  104160                          ERROR   160
 7955  037060  000230                          SPL     0
 7956  037062  005726                          TST     (SP)+
 7957  037064  000002                          RTI                     ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
 7958                                                                  ;FROM THE TEST.
 7959
 7960  037066  000207                RK5ZZ:    RTS     PC              ;THERE WERE NO ERRORS.
 7961
 7962  037070  000000                DRK5T1:   .WORD   0
 7963  037072  000000                DRK5T2:   .WORD   0
 7964
 7965
 7966
 7967                                          ;THIS IS THE UBE DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
 7968                                          ;TEST.
 7969  037074  012737  050050  037442  DRUBE:   MOV     #50050,DUBET1   ;INITIALIZE THE RANDOM DATA
 7970  037102  012737  060060  037444          MOV     #60060,DUBET2   ;GENERATER.
 7971  037110  012737  070070  037250          MOV     #70070,UBEAA3
 7972
 7973  037116  104405                UBEAA:    SAVREG
 7974  037120  004737  037726                  JSR     PC,GETBUF       ;PICK UP A MEMORY BUFFER
```

```
 7975  037124  033776                          .WORD   UBERB
 7976
 7977  037126  013701  033776                  MOV     UBERB,R1        ;COMPUTE THE MEMORY ADDRESS.
 7978  037132  005000                          CLR     R0
 7979  037134  073027  000014                  ASHC    #12.,R0
 7980  037140  010137  037252                  MOV     R1,UBEAA1
 7981  037144  010137  037276                  MOV     R1,UBEDD
 7982  037150  010137  037336                  MOV     R1,UBEII
 7983  037154  010037  037254                  MOV     R0,UBEAA2
 7984  037160  010037  037300                  MOV     R0,UBEEE
 7985  037164  010037  037340                  MOV     R0,UBEJJ
 7986
 7987  037170  000237                          SPL     7
 7988  037172  013737  037442  043056          MOV     DUBET1,$HINUM
 7989  037200  013737  037444  043060          MOV     DUBET2,$LONUM
 7990  037206  004737  042760                  JSR     PC,$RAND
 7991  037212  013737  043056  037442          MOV     $HINUM,DUBET1
 7992  037220  013737  043060  037444          MOV     $LONUM,DUBET2
 7993  037226  000230                          SPL     0
 7994
 7995  037230  013703  037442                  MOV     DUBET1,R3       ;SET THE UNIBUS TESTER DATA REG.
 7996  037234  010337  037334                  MOV     R3,UBEHH
 7997  037240  010337  037274                  MOV     R3,UBECCC
 7998
 7999  037244  104406                          RESREG
 8000
 8001  037246  104420                          WRRAND                  ;FILL THE MEMORY BUFFER WITH
 8002  037250  000000                UBEAA3:   .WORD   0               ;RANDOM DATA.
 8003  037252  000000                UBEAA1:   .WORD   0
 8004  037254  000000                UBEAA2:   .WORD   0
 8005  037256  004000                          .WORD   4000
 8006  037260  005237  037250                  INC     UBEAA3
 8007
 8008  037264  000237                          SPL     7
 8009  037266  104425                          CALUBE                  ;DO A READ MEMORY FUNCTION.
 8010  037270  042543                          .WORD   42543
 8011  037272  000000                UBEBB:    .WORD   0
 8012  037274  000000                UBECCC:   .WORD   0
 8013  037276  000000                UBEDD:    .WORD   0
 8014  037300  000000                UBEEE:    .WORD   0
 8015  037302  010000                          .WORD   10000
 8016  037304  037320                          .WORD   UBEFF
 8017
 8018  037306  004737  037400                  JSR     PC,UBEYY
 8019  037312  005066  000002                  CLR     2(SP)
 8020  037316  000002                          RTI                     ;GO DO SOMETHING ELSE WHILE
 8021                                                                  ;WAITING FOR INTERRUPT.
 8022  037320  004737  037400        UBEFF:    JSR     PC,UBEYY
 8023
 8024  037324  000237                          SPL     7
 8025  037326  104425                          CALUBE                  ;DO A WRITE MEMORY FUNCTION.
 8026  037330  042543                          .WORD   42543
 8027  037332  000000                UBEGG:    .WORD   0
 8028  037334  000000                UBEHH:    .WORD   0
 8029  037336  000000                UBEII:    .WORD   0
 8030  037340  000000                UBEJJ:    .WORD   0
```

```
8031  037342  010000                          .WORD   10000
8032  037344  037360                          .WORD   UBEKK
8033
8034  037346  004737  037400                  JSR     PC,UBEYY
8035  037352  005066  000002                  CLR     2(SP)
8036  037356  000002                          RTI                     ;GO DO SOMETHING ELSE WHILE
8037                                                                  ;WAITING FOR THE INTERRUPT.
8038  037360  004737  037400          UBEKK:  JSR     PC,UBEYY
8039
8040  037364  005337  033740                  DEC     UBECR           ;DECREMENT THE PASS COUNT.
8041  037370  001001                          BNE     UBELL           ;BR IF NOT DONE
8042
8043  037372  000002                          RTI                     ;IF DONE RETURN.
8044  037374  000137  037116          UBELL:  JMP     UBEAA           ;IF NOT DONE DO ANOTHER PASS.
8045
8046  037400  005737  050726          UBEYY:  TST     UBEER1          ;WERE THERE ANY ERRORS?
8047  037404  001415                          BEQ     UBEZZ           ;BR IF NO.
8048
8049  037406  000237                          SPL     7               ;IF THERE WERE REPORT DEVICE FAILURE.
9050  037410  005037  033740                  CLR     UBECR
8051  037414  013737  050730  001634          MOV     UBEER2,$TMP1
8052  037422  013737  050732  001636          MOV     UBEER3,$TMP2
8053  037430  104161                          ERROR   161
8054  037432  005726                          TST     (SP)+
8055  037434  000230                          SPL     0
8056  037436  000002                          RTI                     ;RETURN WITH THIS DRIVER LOCKED OUT.
8057  037440  000207          UBEZZ:  RTS     PC                      ;NO ERRORS CONTINUE.
8058
8059  037442  000000          DUBET1: .WORD   0
8060  037444  000000          DUBET2: .WORD   0
8061
8062
8063
8064                                  ;THIS ROUTINE IS USED TO GENERATE A BUFFER FULL OF RANDOM DATA.
8065                                  ;IT IS CALLED USING THE TRAP TABLE CALL:
8066                                  ;       WRRAND
8067                                  ;       .WORD   HIGHNUM
8068                                  ;       .WORD   LOADRS
8069                                  ;       .WORD   HIGHADRS
8070                                  ;       .WORD   WORDCOUNT
8071                                  ;RET:
8072                                  ;WHERE HIGHNUM IS THE HIGH ORDER PART OF THE NUMBER USED TP PRIME THE
8073                                  ;RANDOM NUMBER GENERATER. THE LOW ORDER PART OF THAT NUMBER IS ASSUMED
8074                                  ;TO BE ZERO. LOADRS AND HIGHADRS IS THE 22 BIT ADDRESS OF THE BUFFER
8075                                  ;IN MEMORY WHICH WILL BE FILLED. WORDCOUNT IS THE NUMBER OF LOCATIONS
8076                                  ;TO BE WRITTEN.
8077  037446  000237          RANDWR: SPL     7
8078  037450  011637  037610                  MOV     (SP),RANDTP
8079  037454  062716  000010                  ADD     #10,(SP)
8080  037460  104405                          SAVREG
8081  037462  013700  037610                  MOV     RANDTP,R0
8082  037466  012001                          MOV     (R0)+,R1
8083  037470  012002                          MOV     (R0)+,R2
8084  037472  012003                          MOV     (R0)+,R3
8085  037474  012004                          MOV     (R0)+,R4
8086  037476  010237  037606                  MOV     R2,RLWT
```

```
 8087  037502  010337  037604                  MOV     R3,RHWT
 8088  037506  010137  043056                  MOV     R1,$HINUM
 8089  037512  005037  043060                  CLR     $LONUM
 8090
 8091  037516  013702  037604          1$:     MOV     RHWT,R2         ;COMPUTE THE VIRTUAL ADDRESS OF THE BUFFER WORD.
 8092  037522  013703  037606                  MOV     RLWT,R3
 8093  037526  073227  177772                  ASHC    #-6,R2
 8094  037532  010337  172354                  MOV     R3,@#KIPAR6
 8095  037536  013702  037606                  MOV     RLWT,R2
 8096  037542  042702  177700                  BIC     #177700,R2
 8097  037546  062702  140000                  ADD     #140000,R2
 8098  037552  004737  042760                  JSR     PC,$RAND
 8099  037556  013712  043056                  MOV     $HINUM,(R2)
 8100  037562  062737  000002  037606          ADD     #2,RLWT
 8101  037570  005537  037604                  ADC     RHWT
 8102  037574  077430                          SOB     R4,1$
 8103
 8104  037576  000230                          SPL     0
 8105  037600  104406                          RESREG
 8106  037602  000002                          RTI
 8107
 8108  037604  000000                  RHWT:   .WORD   0
 8109  037606  000000                  RLWT:   .WORD   0
 8110  037610  000000                  RANDTP: .WORD   0
 8111
 8112                                  ;THIS ROUTINE IS USED TO INITIALIZE THE GET BUFFER ROUTINE.
 8113  037612  012700  033766          GTBINT: MOV     #RS4RB,R0       ;CLEAR ALL THE BUFFER POINTERS.
 8114  037616  012701  000005                  MOV     #5,R1
 8115
 8116  037622  005020                  1$:     CLR     (R0)+
 8117  037624  077102                          SOB     R1,1$
 8118  037626  104412                          SIZE                    ;COMPUTE THE SIZE OF MEMORY.
 8119  037630  000000                  GTBILO: .WORD   0
 8120  037632  000000                  GTBIHI: .WORD   0
 8121  037634  062737  000002  037630          ADD     #2,GTBILO
 8122  037642  005537  037632                  ADC     GTBIHI
 8123  037646  013700  037632                  MOV     GTBIHI,R0       ;COMPUTE THE 2K BLOCK SIZE OF MEMORY.
 8124  037652  013701  037630                  MOV     GTBILO,R1
 8125  037656  073027  177764                  ASHC    #-12.,R0
 8126  037662  010137  037714                  MOV     R1,GTMSIZ
 8127  037666  162701  000011                  SUB     #11,R1
 8128  037672  010137  037716                  MOV     R1,AVMBL
 8129  037676  012737  123456  037720          MOV     #123456,GTRNL
 8130  037704  012737  123456  037722          MOV     #123456,GTRNH
 8131  037712  000207                          RTS     PC
 8132
 8133  037714  000000                  GTMSIZ: .WORD   0
 8134  037716  000000                  AVMBL:  .WORD   0
 8135  037720  000000                  GTRNL:  .WORD   0
 8136  037722  000000                  GTRNH:  .WORD   0
 8137  037724  000000                  GETMP1: .WORD   0
 8138
 8139                                  ;THIS ROUTINE IS CALLED TO ALLOCATE A MEMORY BUFFER OF 2K WORDS LENGTH.
 8140                                  ;IT IS CALLED USING A JSR PC INSTRUCTION FOLLOWED BY THE TABLE ENTRY
 8141                                  ;OF RS4RB TO BE UPDATED.
 8142  037726  000237                  GETBUF: SPL     7               ;LOCK OUT INTERRUPTS.
```

```
8143  037730  011637  037724                  MOV     (SP),GETMP1
8144  037734  062716  000002                  ADD     #2,(SP)         ;PICK UP A POINTER TO THE ARGUMENT
8145                                                                  ;AND UPDATE THE RETURN ADDRESS.
8146  037740  104405                          SAVREG
8147  037742  013737  037720  043060  1$:     MOV     GTRNL,$LONUM
8148  037750  013737  037722  043056          MOV     GTRNH,$HINUM
8149  037756  004737  042760                  JSR     PC,$RAND
8150  037762  013737  043060  037720          MOV     $LONUM,GTRNL
8151  037770  013701  043056                  MOV     $HINUM,R1
8152  037774  010137  037722                  MOV     R1,GTRNH
8153  040000  005000                          CLR     R0
8154  040002  071037  037716                  DIV     AVMBL,R0
8155
8156  040006  012702  033766                  MOV     #RS4RB,R2       ;SEE IF THIS AREA IS ALREADY IN USE.
8157  040012  012703  000005                  MOV     #5,R3
8158  040016  062701  000011                  ADD     #11,R1
8159
8160  040022  020122          2$:     CMP     R1,(R2)+
8161  040024  001746                  BEQ     1$              ;IF IT IS THEN TRY AGAIN.
8162  040026  077303                  SOB     R3,2$
8163
8164  040030  017704  177670                  MOV     @GETMP1,R4      ;OTHERWISE GIVE THIS BUFFER TO THE DRIVER.
8165  040034  010114                          MOV     R1,(R4)
8166  040036  104406                          RESREG
8167  040040  000230                          SPL     0
8168  040042  000207                          RTS     PC
8169
8170
8171  040044  104407          INDONE: RSET
8172
8173
8174
8175
8176                                    ;;*****************************************************************
8177                                    ;*TEST 32       MASS BUS WRITE HIT CYCLE, INVALIDATION TEST
8178                                    ;*
8179                                    ;*THIS IS A TEST OF CACHE INVALIDATION ON MASS BUS CYCLES WHICH ARE
8180                                    ;*WRITE HITS IN THE CACHE. A GROUP OF LOCATIONS IS MADE HITS AND THEN A
8181                                    ;*MASS BUS DEVICE IS CALLED UPON TO DO TRANSFERS, WRITES TO THOSE
8182                                    ;*LOCATIONS. THOSE WRITES SHOULD THUS BE INVALIDATED.
8183                                    ;*
8184                                    ;;*****************************************************************
8185  040046  000004          TST32:  SCOPE
8186                                                          ;SET THE SKAD REGISTER
8187  040050  012737  041174  043632          MOV     #$EOP,SKAD      ;IN CASE THE TEST ABORTS.
8188
8189  040056  104407                          RSET
8190  040060  113737  001502  001632          MOVB    $TSTNM,$TMP0
8191  040066  004737  045046                  JSR     PC,SIZDEV       ;DETERMINE WHAT DEVICES ARE AVAILABLE.
8192  040072  113737  045440  040606          MOVB    RS4DFL,RS4FT
8193  040100  113737  045441  040607          MOVB    RP4DFL,RP4FT
8194  040106  113737  045442  040610          MOVB    RH4DFL,RH4FT
8195
8196  040114  000137  040724          NN1:    JMP     NNDEV           ;GO COMPUTE THE DRIVE NUMBERS.
8197
8198  040120  005037  040604          NN2:    CLR     NNGRPF          ;FLAG WHICH DESIGNATES WHICH GOUP IS BEING
```

```
8199  040124  012737  000044  040602          MOV     #S1M0,NNGRM     ;TESTED ON THIS PASS.
8200  040132  012737  000030  040600          MOV     #S0M1,NNGRS     ;TEST GROUP ZERO FIRST.
8201
8202  040140  004737  040614          NN3:    JSR     PC,NNSTUP       ;GO MAKE THE TEST ADDRESSES HITS
8203  040144  004777  000426                  JSR     PC,@NNUD                ;USE THE FIRST DEVICE.
8204
8205
8206  040150  012700  140000                  MOV     #TESTR1,R0
8207  040154  012701  000400                  MOV     #256.,R1        ;MAKE SURE THOSE ADDRESSES ARE MISSES.
8208
8209  040160  005710                  1$:     TST     (R0)
8210  040162  032737  000010  177752          BIT     #10,@#HITMIS
8211  040170  001430                          BEQ     2$
8212
8213  040172  013737  040604  001634          MOV     NNGRPF,$TMP1    ;GOT A HIT REPORT FAILURE.
8214  040200  010037  001636                  MOV     R0,$TMP2
8215  040204  005037  001640                  CLR     $TMP3
8216  040210  023727  040576  040412          CMP     NNUD,#NNRS4     ;WAS THE RS4 DOING THE TRANSFER?
8217  040216  001003                          BNE     11$             ;BRANCH IF NOT.
8218  040220  104151                          ERROR   151
8219  040222  000137  040260                  JMP     NN5
8220  040226  023727  040576  040504  11$:    CMP     NNUD,#NNRP4     ;WAS IT THE RP4?
8221  040234  001003                          BNE     12$
8222  040236  104152                          ERROR   152
8223  040240  000137  040260                  JMP     NN5
8224  040244  104153                  12$:    ERROR   153
8225  040246  000137  040260                  JMP     NN5
8226
8227  040252  062700  000004          2$:     ADD     #4,R0
8228  040256  077140                          SOB     R1,1$
8229
8230  040260  005237  040604          NN5:    INC     NNGRPF          ;TESTED BOTH GROUPS?
8231  040264  022737  000002  040604          CMP     #2,NNGRPF
8232  040272  001410                          BEQ     NN6             ;BRANCH IF YES.
8233  040274  012737  000044  040600          MOV     #S1M0,NNGRS     ;IF NOT GO BACK AND TEST GROUP ONE.
8234  040302  012737  000030  040602          MOV     #S0M1,NNGRM
8235  040310  000137  040140                  JMP     NN3
8236
8237  040314  000137  041172          NN6:    JMP     NNDONE
8238
8239  040320  104423                  NNRH4:  CALRH4          ;THIS IS THE CALL TO READ THE MASS BUS TESTER.
8240  040322     071                          .BYTE   71
8241  040323     000                  NNRH4U: .BYTE   0
8242  040324  052525                          .WORD   52525
8243  040326  000000                          .WORD   0
8244  040330  140000                          .WORD   TESTR1
8245  040332  000000                          .WORD   0
8246  040334  001000                          .WORD   512.
8247  040336  040350                          .WORD   2$
8248
8249  040340  005737  051460          1$:     TST     RH4ER1          ;ANY DEVICE ERRORS?
8250  040344  100401                          BMI     2$              ;BRANCH IF YES.
8251  040346  000207                          RTS     PC              ;IF NOT RETURN.
8252
8253  040350  013737  051462  001634  2$:     MOV     RH4ER2,$TMP1    ;REPORT DEVICE ERROR.
8254  040356  013737  051464  001636          MOV     RH4ER3,$TMP2
```

```
8255  040364  013737  051466  001640          MOV     RH4ER4,$TMP3
8256  040372  005726                          TST     (SP)+
8257  040374  104156                          ERROR   156
8258  040376  105037  045442                  CLRB    RH4DFL
8259  040402  105037  040610                  CLRB    RH4FT
8260  040406  000137  040114                  JMP     NN1
8261
8262  040412  104421                  NNRS4:  CALRS4                  ;THIS IS A CALL TO DO AN RS4 READ.
8263  040414     071                          .BYTE   71
8264  040415     000                  NNRS4U: .BYTE   0
8265  040416  000000                          .WORD   0
8266  040420  000000                          .WORD   0
8267  040422  140000                          .WORD   TESTR1
8268  040424  000000                          .WORD   0
8269  040426  001000                          .WORD   512.
8270  040430  040442                          .WORD   2$
8271
8272  040432  005737  046756          1$:     TST     RS4ER1          ;SEE IF THERE WERE DEVICE ERRORS.
8273  040436  100401                          BMI     2$              ;BR IF YES.
8274  040440  000207                          RTS     PC
8275
8276  040442  013737  046760  001634  2$:     MOV     RS4ER2,$TMP1
8277  040450  013737  046762  001636          MOV     RS4ER3,$TMP2
8278  040456  013737  046764  001640          MOV     RS4ER4,$TMP3
8279  040464  005726                          TST     (SP)+
8280  040466  104154                          ERROR   154
8281  040470  105037  045440                  CLRB    RS4DFL
8282  040474  105037  040606                  CLRB    RS4FT
8283  040500  000137  040114                  JMP     NN1
8284
8285  040504  104422                  NNRP4:  CALRP4                  ;THIS IS A CALL TO DO AN RP4 READ.
8286  040506     071                          .BYTE   71
8287  040507     000                  NNRP4U: .BYTE   0
8288  040510  000000                          .WORD   0
8289  040512  000000                          .WORD   0
8290  040514  140000                          .WORD   TESTR1
8291  040516  000000                          .WORD   0
8292  040520  001000                          .WORD   512.
8293  040522  040534                          .WORD   2$
8294
8295  040524  005737  046014          1$:     TST     RP4ER1          ;WERE THERE ANY DEVICE ERRORS?
8296  040530  100401                          BMI     2$
8297  040532  000207                          RTS     PC
8298
8299  040534  013737  046016  001634  2$:     MOV     RP4ER2,$TMP1
8300  040542  013737  046020  001636          MOV     RP4ER3,$TMP2
8301  040550  013737  046022  001640          MOV     RP4ER4,$TMP3
8302  040556  005726                          TST     (SP)+
8303  040560  104155                          ERROR   155
8304  040562  105037  045441                  CLRB    RP4DFL
8305  040566  105037  040607                  CLRB    RP4FT
8306  040572  000137  040114                  JMP     NN1
8307
8308  040576  000000                  NNUD:   .WORD   0
8309
8310  040600  000000                  NNGRS:  .WORD   0
```

```
8311  040602  000000                  NNGRM:  .WORD   0
8312  040604  000000                  NNGRPF: .WORD   0
8313
8314                                  ;THIS ROUTINE IS CALLED TO MAKE THE ADDRESSES IN TESTR1
8315                                  ;HITS PRIOR TO CALLING FOR THE MB DEVICE TO DO TRANSFERS.
8316  040606     000                  RS4FT:  .BYTE   0
8317  040607     000                  RP4FT:  .BYTE   0
8318  040610     000                  RH4FT:  .BYTE   0
8319  040611     000                  RK5FT:  .BYTE   0
8320  040612     000                  UBEFT:  .BYTE   0
8321          040614                        .EVEN
8322
8323  040614  104405                  NNSTUP: SAVREG
8324  040616  012700  040614                  MOV     #NNSTUP,R0      ;MAKE THIS CODE HITS IN THE
8325  040622  012701  001000                  MOV     #512.,R1        ;GROUP NOT BEING TESTED.
8326  040626  012702  142000                  MOV     #TESTR2,R2
8327
8328  040632  013737  040602  177746  1$:     MOV     NNGRM,@#CONTRL
8329  040640  005720                        TST     (R0)+
8330  040642  013737  040600  177746        MOV     NNGRS,@#CONTRL
8331  040650  005722                        TST     (R2)+
8332  040652  077111                        SOB     R1,1$
8333
8334  040654  013700  040600          2$:     MOV     NNGRS,R0
8335  040660  042700  000014                  BIC     #14,R0
8336  040664  010037  177746                  MOV     R0,@#CONTRL
8337  040670  012701  140000                  MOV     #TESTR1,R1
8338  040674  012702  001000                  MOV     #512.,R2
8339  040700  005721                  3$:     TST     (R1)+
8340  040702  077202                        SOB     R2,3$
8341  040704  013700  040602                  MOV     NNGRM,R0
8342  040710  042700  000014                  BIC     #14,R0
8343  040714  010037  177746                  MOV     R0,@#CONTRL
8344  040720  104406                        RESREG
8345  040722  000207                        RTS     PC
8346
8347
8348                                  ;SEE WHAT DEVICE TO USE NEXT.
8349  040724  000240                  NNDEV:  NOP
8350  040726  000240                          NOP
8351  040730  005037  040576                  CLR     NNUD
8352  040734  113700  040606                  MOVB    RS4FT,R0                ;IS THERE AN RS4 DRIVE.
8353  040740  001430                          BEQ     NND2                    ;BR IS NOT
8354
8355  040742  000240                  NND0:   NOP
8356  040744  012701  000001                  MOV     #1,R1                   ;FIND OUT WHAT DRIVE  NUMBER IT IS.
8357  040750  012737  040412  040576        MOV     #NNRS4,NNUD
8358  040756  005002                        CLR     R2
8359  040760  012703  000010                  MOV     #10,R3
8360  040764  000240                  1$:     NOP
8361  040766  030100                          BIT     R1,R0
8362  040770  001406                          BEQ     2$
8363  040772  140137  040606                  BICB    R1,RS4FT                ;FOUND IT.
8364  040776  110237  040415                  MOVB    R2,NNRS4U
8365  041002  000137  040120                  JMP     NN2
8366  041006  005202                  2$:     INC     R2
```

```
8367  041010  006301                          ASL     R1
8368  041012  077314                          SOB     R3,1$                   ;KEEP LOOKING.
8369
8370  041014  104000                          ERROR   0
8371  041016  105037  040606                  CLRB    RS4FT
8372
8373  041022  000240                  NND2:   NOP
8374  041024  113700  040607                  MOVB    RP4FT,R0                ;IS THERE AN RP04 DRIVE.
8375  041030  001426                          BEQ     NND3                    ;BR IF NO
8376  041032  012701  000001                  MOV     #1,R1
8377  041036  012737  040504  040576          MOV     #NNRP4,NNUD
8378  041044  005002                          CLR     R2
8379  041046  012703  000010                  MOV     #10,R3
8380  041052  030100                  1$:     BIT     R1,R0
8381  041054  001406                          BEQ     2$
8382  041056  140137  040606                  BICB    R1,RS4FT
8383  041062  110237  040507                  MOVB    R2,NNRP4U
8384  041066  000137  040120                  JMP     NN2
8385  041072  005202                  2$:     INC     R2
8386  041074  006301                          ASL     R1
8387  041076  077313                          SOB     R3,1$
8388  041100  104000                          ERROR   0
8389  041102  105037  040607                  CLRB    RP4FT
8390
8391  041106  000240                  NND3:   NOP
8392  041110  113700  040610                  MOVB    RH4FT,R0                ;IS THERE A MASS BUS TESTER.
8393  041114  001426                          BEQ     NNDONE
8394  041116  012701  000001                  MOV     #1,R1
8395  041122  012737  040320  040576          MOV     #NNRH4,NNUD
8396  041130  005002                          CLR     R2
8397  041132  012703  000010                  MOV     #10,R3
8398  041136  030100                  1$:     BIT     R1,R0
8399  041140  001406                          BEQ     2$
8400  041142  140137  040610                  BICB    R1,RH4FT
8401  041146  110237  040323                  MOVB    R2,NNRH4U
8402  041152  000137  040120                  JMP     NN2
8403  041156  005202                  2$:     INC     R2
8404  041160  006301                          ASL     R1
8405  041162  077313                          SOB     R3,1$
8406  041164  104000                          ERROR   0
8407  041166  105037  040610                  CLRB    RH4FT
8408  041172  104407                  NNDONE: RSET
8409
8410
8411                                  .SBTTL  END OF PASS ROUTINE
8412
8413                                  ;;**************************************************************
8414                                  ;*INCREMENT THE PASS NUMBER ($PASS)
8415                                  ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
8416                                  ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
8417                                  ;*IF THERES A MONITOR GO TO IT
8418                                  ;*IF THERE ISN'T JUMP TO LOOP
8419
8420  041174                          $EOP:
8421  041174  000004                          SCOPE
8422  041176  005037  001502                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
```

```
 8423  041202  005037  001702                    CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 8424  041206  005237  001500                    INC     $PASS           ;;INCREMENT THE PASS NUMBER
 8425  041212  042737  100000  001500            BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 8426  041220  005327                            DEC     (PC)+           ;;LOOP?
 8427  041222  000001                $EOPCT:     .WORD   1
 8428  041224  003031                            BGT     $DOAGN          ;;YES
 8429  041226  012737                            MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 8430  041230  000001                $ENDCT:     .WORD   1
 8431  041232  041222                            $EOPCT
 8432  041234  104400  041317                    TYPE    ,$ENDMG         ;;TYPE "END PASS #"
 8433  041240  013746  001500                    MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 8434  041244  104404                      —     TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 8435  041246  104400  041314                    TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
 8436  041252  013700  000042          $GET42:   MOV     @#42,R0         ;;GET MONITOR ADDRESS
 8437  041256  001414                            BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 8438  041260  012703  125252                    MOV     #125252,R3
 8439  041264  004737  044142                    JSR     PC,CHAINQ
 8440  041270  013700  000042                    MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
 8441  041274  001405                            BEQ     $DOAGN          ;;RETURN ADDRESS
 8442  041276  000005                            RESET                   ;;CLEAR THE WORLD
 8443  041300  004710                $ENDAD:     JSR     PC,(R0)         ;;GO TO MONITOR
 8444  041302  000240                            NOP                     ;;SAVE ROOM
 8445  041304  000240                            NOP                     ;;FOR
 8446  041306  000240                            NOP                     ;;ACT11
 8447  041310                        $DOAGN:
 8448  041310  000137                            JMP     @(PC)+          ;;RETURN
 8449  041312  004330                $RTNAD:     .WORD   LOOP
 8450  041314     377     377     000  $ENULL:   .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 8451  041317     015  042412  042116  $ENDMG:   .ASCIZ  <15><12>/END PASS #/
 8452  041324  050040  051501  020123
 8453  041332  000043
 8454
 8455                                    .SBTTL  SCOPE HANDLER ROUTINE
 8456
 8457                                    ;;**************************************************************
 8458                                    ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 8459                                    ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 8460                                    ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 8461                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 8462                                    ;*SW14=1        LOOP ON TEST
 8463                                    ;*SW11=1        INHIBIT ITERATIONS
 8464                                    ;*SW09=1        LOOP ON ERROR
 8465                                    ;*SW08=1        LOOP ON TEST IN SWR<6:0>
 8466                                    ;*CALL
 8467                                    ;*      SCOPE           ;;SCOPE=IOT
 8468
 8469  041334                            $SCOPE:
 8470  041334  032777  040000  140176  1$:       BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 8471  041342  001114                            BNE     $OVER           ;;YES IF SW14=1
 8472                                    ;*****START OF CODE FOR THE XOR TESTER*****
 8473  041344  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
 8474                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 8475  041346  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 8476  041352  012737  041372  000004            MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 8477  041360  005737  177060                    TST     @#177060        ;;TIME OUT ON XOR?
 8478  041364  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
```

```
 9479  041370  000466                          BR      $SVLAD          ;;GO TO THE NEXT TEST
 9480  041372  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 9481  041374  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9482  041400  000426                          BR      7$              ;;LOOP ON THE PRESENT TEST
 9483  041402                          6$:;*****END OF CODE FOR THE XOR TESTER*****
 9484  041402  032777  000400  140130          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
 9485  041410  001407                          BEQ     2$              ;;BR IF NO
 9486  041412  017746  140122                  MOV     @SWR,-(SP)      ;;SET DESIRED TEST NUM. FROM SWR
 9487  041416  042716  000200                  BIC     #SSWRMK,(SP)    ;;STRIP AWAY UNDESIRED BITS
 9488  041422  122637  001502                  CMPB    (SP)+,$TSTNM    ;;ON THE RIGHT TEST?
 9489  041426  001462                          BEQ     $OVER           ;;BR IF YES
 9490  041430  105737  001503          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 9491  041434  001421                          BEQ     3$              ;;BR IF NO
 9492  041436  123737  001515  001503          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 9493  041444  101015                          BHI     3$              ;;BR IF NO
 9494  041446  032777  001000  140064          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 9495  041454  001404                          BEQ     4$              ;;BR IF NO
 9496  041456  013737  001510  001506  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 9497  041464  000443                          BR      $OVER
 9498  041466  105037  001503          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 9499  041472  005037  001702                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 8500  041476  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
 8501  041500  032777  004000  140032  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 8502  041506  001011                          BNE     1$              ;;BR IF YES
 8503  041510  005737  001500                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 8504  041514  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
 8505  041516  005237  001504                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
 8506  041522  023737  001702  001504          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 8507  041530  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 8508  041532  012737  000001  001504  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 8509  041540  013737  041610  001702          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 8510  041546  105237  001502          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 8511  041552  011637  001506                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 8512  041556  011637  001510                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 8513  041562  005037  001704                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 8514  041566  112737  000001  001515          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 8515  041574  013777  001502  137740  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 8516  041502  013716  001506                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 8517  041606  000002                          RTI                     ;;FIXES PS
 8518  041610  000001                  $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
 8519
 8520                                  .SBTTL  ERROR HANDLER ROUTINE
 8521
 8522                                  ;;**************************************************************
 8523                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 8524                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 8525                                  ;*AND GO TO ERTYPE ON ERROR
 8526                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 8527                                  ;*SW15=1         HALT ON ERROR
 8528                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
 8529                                  ;*SW10=1         BELL ON ERROR
 8530                                  ;*SW09=1         LOOP ON ERROR
 8531                                  ;*CALL
 8532                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 8533
 8534  041612                          $ERROR:
```

```
8535  041612  105237  001503          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
8536  041616  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
8537  041620  013777  001502  137714          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
8538  041626  032777  002000  137704          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
8539  041634  001402                          BEQ     1$              ;;NO - SKIP
8540  041636  104400  001706                  TYPE    ,$BELL          ;;RING BELL
8541  041642  005237  001512          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
8542  041646  011637  001516                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
8543  041652  162737  000002  001516          SUB     #2,$ERRPC
8544  041660  116737  137632  001514          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
8545  041666  032777  020000  137644          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
8546  041674  001004                          BNE     20$             ;;SKIP TYPEOUTS
8547  041676  004737  044334                  JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
8548  041702  104400  001713                  TYPE    ,$CRLF
8549  041706                          20$:
8550  041706  005777  137626          2$:     TST     @SWR            ;;HALT ON ERROR
8551  041712  100001                          BPL     3$              ;;SKIP IF CONTINUE
8552  041714  000000                          HALT                    ;;HALT ON ERROR!
8553  041716  032777  001000  137614  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
8554  041724  001402                          BEQ     4$              ;;BR IF NO
8555  041726  013716  001510                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
8556  041732  005737  001704          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
8557  041736  001402                          BEQ     5$              ;;BR IF NONE
8558  041740  013716  001704                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
8559  041744                          5$:
8560  041744  022737  041300  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
8561  041752  001001                          BNE     6$              ;;BRANCH IF NO
8562  041754  000000                          HALT                    ;;YES
8563  041756                          6$:
8564  041756  012737  177777  177744          MOV     #-1,@#MEMERR
8565  041764  005037  177766                  CLR     @#CPUERR
8566  041770  000002                          RTI
8567
8568                                  .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
8569
8570                                  ;;**************************************************************
8571                                  ;*SAVE R0-R5
8572                                  ;*CALL:
8573                                  ;*      SAVREG
8574                                  ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
8575                                  ;*
8576                                  ;*TOP---(+16)
8577                                  ;* +2---(+18)
8578                                  ;* +4---R5
8579                                  ;* +6---R4
8580                                  ;* +8---R3
8581                                  ;*+10---R2
8582                                  ;*+12---R1
8583                                  ;*+14---R0
8584
8585  041772                          $SAVREG:
8586  041772  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8587  041774  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8588  041776  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8589  042000  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8590  042002  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
```

```
8591  042004  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8592  042006  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
8593  042012  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
8594  042016  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
8595  042022  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
8596  042026  000002                          RTI
8597
8598                                  ;*RESTORE R0-R5
8599                                  ;*CALL:
8600                                  ;*      RESREG
8601  042030                          $RESREG:
8602  042030  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
8603  042034  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
8604  042040  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
8605  042044  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
8606  042050  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
8607  042052  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
8608  042054  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
8609  042056  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
8610  042060  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
8611  042062  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
8612  042064  000002                          RTI
8613
8614                                  .SBTTL  TYPE ROUTINE
8615
8616                                  ;;**************************************************************
8617                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
8618                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
8619                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
8620                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
8621                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
8622                                  ;*
8623                                  ;*CALL:
8624                                  ;*1) USING A TRAP INSTRUCTION
8625                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
8626                                  ;*OR
8627                                  ;*      TYPE
8628                                  ;*      MESADR
8629                                  ;*
8630
8631  042066  105737  001557          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
8632  042072  100002                          BPL     1$              ;;BR IF YES
8633  042074  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
8634  042076  000407                          BR      3$              ;;LEAVE
8635  042100  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
8636  042102  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
8637  042106  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
8638  042110  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
8639  042112  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
8640  042114  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
8641  042116  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
8642  042122  000002                          RTI                     ;;RETURN
8643  042124  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
8644  042130  001430                          BEQ     8$
8645  042132  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
8646  042136  001006                          BNE     5$
```

```
 8647  042140  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
 8648  042142  104400                          TYPE                    ;;TYPE A CR AND LF
 8649  042144  001713                          $CRLF
 8650  042146  105037  042302                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 8651  042152  000755                          BR      2$              ;;GET NEXT CHARACTER
 8652  042154  004737  042236          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 8653  042160  123726  001556          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 8654  042164  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
 8655  042166  013746  001554                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 8656                                                                  ;;AND THE NULL CHAR.
 8657  042172  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 8658  042176  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 8659  042200  004737  042236                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
 8660  042204  105337  042302                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 8661  042210  000770                          BR      7$              ;;LOOP
 8662
 8663                                  ;HORIZONTAL TAB PROCESSOR
 8664
 8665  042212  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 8666  042216  004737  042236          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 8667  042222  132737  000007  042302          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 8668  042230  001372                          BNE     9$              ;;TAB STOP
 8669  042232  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
 8670  042234  000724                          BR      2$              ;;GET NEXT CHARACTER
 8671  042236  105777  137306          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 8672  042242  100375                          BPL     $TYPEC
 8673  042244  116677  000002  137300          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 8674  042252  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 8675  042260  001003                          BNE     1$              ;;BRANCH IF NO
 8676  042262  105037  042302                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 8677  042266  000406                          BR      $TYPEX          ;;EXIT
 8678  042270  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 8679  042276  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
 8680  042300  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
 8681  042302  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 8682  042304  000207                  $TYPEX: RTS     PC
 8683
 8684
 8685                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 8686
 8687                                  ;;**************************************************************
 8688                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 8689                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 8690                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 8691                                  ;*CALL:
 8692                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 8693                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
 8694                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 8695                                  ;*      .BYTE   M               ;;M=1 OR 0
 8696                                  ;*                                      ;;1=TYPE LEADING ZEROS
 8697                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
 8698                                  ;*
 8699                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 8700                                  ;*$TYPOS OR $TYPOC
 8701                                  ;*CALL:
 8702                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
```

```
 8703                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
 8704                                   ;*
 8705                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 8706                                   ;*CALL:
 8707                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 8708                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
 8709
 8710  042306  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 8711  042312  116637  000001  042531           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 8712  042320  116637  042533                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 8713  042324  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 8714  042330  000406                           BR      $TYPON
 8715  042332  112737  000001  042531  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 8716  042340  112737  000006  042533           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 8717  042346  112737  000005  042530  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 8718  042354  010346                           MOV     R3,-(SP)        ;;SAVE R3
 8719  042356  010446                           MOV     R4,-(SP)        ;;SAVE R4
 8720  042360  010546                           MOV     R5,-(SP)        ;;SAVE R5
 8721  042362  113704  042533                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 8722  042366  005404                           NEG     R4
 8723  042370  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 8724  042374  110437  042532                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 8725  042400  113704  042531                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 8726  042404  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 8727  042410  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
 8728  042412  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 8729  042414  000404                           BR      3$              ;;GO DO MSB
 8730  042416  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
 8731  042420  006105                           ROL     R5
 8732  042422  006105                           ROL     R5
 8733  042424  010503                           MOV     R5,R3
 8734  042426  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 8735  042430  105337  042532                   DECB    $OMODE          ;;TYPE THIS DIGIT?
 8736  042434  100016                           BPL     7$              ;;BR IF NO
 8737  042436  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
 8738  042442  001002                           BNE     4$              ;;TEST FOR 0
 8739  042444  005704                           TST     R4              ;;SUPPRESS THIS 0?
 8740  042446  001403                           BEQ     5$              ;;BR IF YES
 8741  042450  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 8742  042452  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 8743  042456  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 8744  042462  110337  042526                   MOVB    R3,8$           ;;SAVE FOR TYPING
 8745  042466  104400  042526                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
 8746  042472  105337  042530           7$:     DECB    $OCNT           ;;COUNT BY 1
 8747  042476  003347                           BGT     2$              ;;BR IF MORE TO DO
 8748  042500  002402                           BLT     6$              ;;BR IF DONE
 8749  042502  005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 8750  042504  000744                           BR      2$              ;;GO DO THE LAST DIGIT
 8751  042506  012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
 8752  042510  012604                           MOV     (SP)+,R4        ;;RESTORE R4
 8753  042512  012603                           MOV     (SP)+,R3        ;;RESTORE R3
 8754  042514  016666  000002  000004           MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 8755  042522  012616                           MOV     (SP)+,(SP)
 8756  042524  000002                           RTI                     ;;RETURN
 8757  042526     000                   8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 8758  042527     000                           .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
```

```
 8759  042530  000                    $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 8760  042531  000                    $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 8761  042532  000000                 $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 8762
 8763                                 .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 8764
 8765                                 ;;**************************************************************
 8766                                 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 8767                                 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 8768                                 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 8769                                 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 8770                                 ;*REPLACED WITH SPACES.
 8771                                 ;*CALL:
 8772                                 ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 8773                                 ;*      TYPDS                   ;;GO TO THE ROUTINE
 8774
 8775  042534                         $TYPDS:
 8776  042534  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8777  042536  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8778  042540  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8779  042542  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 8780  042544  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 8781  042546  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 8782  042552  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 8783  042556  100004                         BPL     1$              ;;BR IF INPUT IS POS.
 8784  042560  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 8785  042562  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 8786  042570  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 8787  042572  012703  042750                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 8788  042576  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 8789  042602  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 8790  042604  016001  042740                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 8791  042610  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 8792  042612  002402                         BLT     4$              ;;BR IF DONE
 8793  042614  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 8794  042616  000774                         BR      3$
 8795  042620  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 8796  042622  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
 8797  042624  001002                         BNE     5$              ;;FALL THROUGH IF 0
 8798  042626  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 8799  042630  100407                         BMI     7$              ;;BR IF YES
 8800  042632  106316                 5$:     ASLB    (SP)            ;;MSD?
 8801  042634  103003                         BCC     6$              ;;BR IF NO
 8802  042636  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 8803  042644  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 8804  042650  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 8805  042654  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 8806  042656  005720                         TST     (R0)+           ;;JUST INCREMENTING
 8807  042660  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
 8808  042664  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
 8809  042666  003002                         BGT     8$              ;;GO TO EXIT
 8810  042670  010502                         MOV     R5,R2           ;;GET THE LSD
 8811  042672  000764                         BR      6$              ;;GO CHANGE TO ASCII
 8812  042674  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 8813  042676  100003                         BPL     9$              ;;BR IF NO
 8814  042700  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
```

```
 8815  042706  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 8816  042710  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 8817  042712  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 8818  042714  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 8819  042716  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8820  042720  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 8821  042722  104400  042750                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 8822  042726  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 8823  042734  012616                          MOV     (SP)+,(SP)
 8824  042736  000002                          RTI                     ;;RETURN TO USER
 8825  042740  023420                  $DTBL:  10000.
 8826  042742  001750                          1000.
 8827  042744  000144                          100.
 8828  042746  000C12                          10.
 8829  042750  000004                  $DBLK:  .BLKW   4
 8830
 8831                                  .SBTTL  RANDOM NUMBER GENERATOR ROUTINE
 8832
 8833                                  ;;**********************************************************************
 8834                                  ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
 8835                                  ;*WITH A RANGE OF 0 TO 2(+33)-1.
 8836                                  ;*CALL:
 8837                                  ;*      JSR     PC,$RAND        ;;CALL THE ROUTINE
 8838                                  ;*      RETURN                  ;;RETURN HERE THE RANDOM
 8839                                  ;*                              ;;NUMBER WILL BE IN
 8840                                  ;*                              ;;$HINUM,$LONUM
 8841
 8842  042760                          $RAND:
 8843  042760  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8844  042762  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8845  042764  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8846  042766  013700  043060                  MOV     $LONUM,R0       ;;SET R0 WITH LOW
 8847  042772  013701  043056                  MOV     $HINUM,R1       ;;SET R1 WITH HIGH
 8848  042776  012702  177771                  MOV     #-7,R2          ;;SET SHIFT COUNT
 8849  043002  006300                  1$:     ASL     R0              ;;SHIFT R0 LEFT AND
 8850  043004  006101                          ROL     R1              ;;ROTATE CARRY INTO R1 AND
 8851  043006  005202                          INC     R2              ;;CHECK FOR DONE
 8852  043010  001374                          BNE     1$              ;;CONTINUE SHIFT LOOP
 8853  043012  063700  043060                  ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
 8854  043016  005501                          ADC     R1              ;;PROPOGATE CARRY
 8855  043020  063701  043056                  ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
 8856  043024  062700  001057                  ADD     #1057,R0        ;;ADD LOW CONSTANT
 8857  043030  005501                          ADC     R1              ;;PROPOGATE CARRY
 8858  043032  062701  047401                  ADD     #47401,R1       ;;ADD HIGH CONSTANT
 8859  043036  010037  043060                  MOV     R0,$LONUM       ;;SAVE R0
 8860  043042  010137  043056                  MOV     R1,$HINUM       ;;SAVE R1
 8861  043046  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 8862  043050  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8863  043052  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 8864  043054  000207                          RTS     PC              ;;RETURN
 8865  043056  176543                  $HINUM: .WORD   176543
 8866  043060  123456                  $LONUM: .WORD   123456
 8867
 8868                                  .SBTTL  TRAP DECODER
 8869
 8870                                  ;;**********************************************************************
```

```
                                ;*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.

043062  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
043064  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
043070  005740                          TST     -(R0)           ;;BACKUP BY 2
043072  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
043074  006300                          ASL     R0              ;;POSITION FOR INDEXING
043076  016000  043104                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
043102  000200                          RTS     R0              ;;GO TO ROUTINE

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
043104                          $TRPAD:
043104  042066                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
043106  042332                          $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
043110  042306                          $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
043112  042346                          $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
043114  042534                          $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)

043116  041772                          $SAVREG ;;CALL=SAVREG   TRAP+5(104405)  SAVE R0-R5 ROUTINE
043120  042030                          $RESREG ;;CALL=RESREG   TRAP+6(104406)  RESTORE R0-R5 ROUTINE

043122  043634                          CLEAN   ;;CALL=RSET     TRAP+7(104407)  GO RESET ALL REGISTERS.
043124  043604                          ABORTT  ;;CALL=SKIPT    TRAP+10(104410) THIS WILL SKIP TO THE NEXT TEST
043126  044236                          MMCES   ;;CALL=MMSKIP   TRAP+11(104411) IF SWITCH # IS ON SKIP TO THE NEXT TEST
043130  044260                          MSIZER  ;;CALL=SIZE     TRAP+12(104412) DETERMINE THE HIGHEST ADDRESS IN MEMORY
043132  043724                          SKBADR  ;;CALL=SKPBAD   TRAP+13(104413) SKIP TEST IF ERROR ADDRESS REGISTER IS I
043134  043750                          SKBERR  ;;CALL=SKPBER   TRAP+14(104414) SKIP TEST IF ERROR REGISTER IS INOPERATI
043136  043766                          SKBCNR  ;;CALL=SKPBCN   TRAP+15(104415) SKIP TEST IF CONTROL REGISTER IS INOPERA
043140  044004                          SKBMNR  ;;CALL=SKPBMN   TRAP+16(104416) SKIP TEST IF MAINTENANCE REGISTER IS INO
043142  044022                          SKBHMR  ;;CALL=SKPBHM   TRAP+17(104417) SKIP TEST IF HIT/MISS REGISTER IS IN OPE
043144  037446                          RANDWR  ;;CALL=WRRAND   TRAP+20(104420) FILL BUFFER WITH RANDOM SEQUENCE

043146  047020                          RS4HAN  ;;CALL=CALRS4   TRAP+21(104421) DO RS04 FUNCTION
043150  046056                          RP4HAN  ;;CALL=CALRP4   TRAP+22(104422) DO RP04 FUNCTION
043152  051522                          RH4HAN  ;;CALL=CALRH4   TRAP+23(104423) DO MBT FUNCTION
043154  047754                          RK5HAN  ;;CALL=CALRK5   TRAP+24(104424) DO RK05 FUNCTION
043156  050770                          UBEHAN  ;;CALL=CALUBE   TRAP+25(104425) DO UBE FUNCTION
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**************************************************************
                                ;POWER DOWN ROUTINE
043160  012737  043324  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
043166  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
043174  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043176  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043200  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
```

```
 8927  043202  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 8928  043204  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 8929  043206  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 8930  043210  017746  136324                MOV     @#SWR,-(SP)     ;;PUSH @SWR ON STACK
 8931  043214  010637  043330                MOV     SP,$SAVR6       ;;SAVE SP
 8932  043220  012737  043232  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 8933  043226  000000                        HALT
 8934  043230  000776                        BR      .-2             ;;HANG UP
 8935
 8936                                  ;;*****************************************************************
 8937                                  ;POWER UP ROUTINE
 8938  043232  012737  043324  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 8939  043240  013706  043330                MOV     $SAVR6,SP       ;;GET SP
 8940  043244  005037  043330                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 8941  043250  005237  043330          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
 8942  043254  001375                        BNE     1$              ;;OF  WORD
 8943  043256  012677  136256                MOV     (SP)+,@#SWR     ;;POP STACK INTO @SWR
 8944  043262  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 8945  043264  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 8946  043266  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 8947  043270  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 8948  043272  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 8949  043274  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 8950  043276  012737  043160  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 8951  043304  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
 8952  043312  104400                        TYPE                    ;;REPORT THE POWER FAILURE
 8953  043314  052413                $PWRMG: .WORD   POWERM          ;;POWER FAIL MESSAGE POINTER
 8954  043316  012716                        MOV     (PC)+,(SP)      ;;RESTART AT START
 8955  043320  003752                $PWRAD: .WORD   START           ;;RESTART ADDRESS
 8956  043322  000002                        RTI
 8957  043324  000000                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
 8958  043326  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 8959  043330  000000                $SAVR6: 0                       ;;PUT THE SP HERE
 8960
 8961                                  .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
 8962
 8963                                  ;;*****************************************************************
 8964                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
 8965                                  ;*UNSIGNED OCTAL ASCIZ NUMBER.
 8966                                  ;*CALL
 8967                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 8968                                  ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
 8969                                  ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
 8970
 8971
 8972  043332  104405                $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
 8973  043334  016601  000002                MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
 8974  043340  012705  043451                MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
 8975  043344  012704  000014                MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
 8976  043350  012703  177770                MOV     #^C7,R3         ;;MASK
 8977  043354  012100                        MOV     (R1)+,R0        ;;LOWER WORD
 8978  043356  012101                        MOV     (R1)+,R1        ;;HIGH WORD
 8979  043360  005002                        CLR     R2              ;;TERMINATOR
 8980  043362  110245                1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
 8981  043364  010002                        MOV     R0,R2           ;;GET THIS DIGIT
 8982  043366  005304                        DEC     R4              ;;COUNT THIS CHARACTER
```

```
 8983  043370  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
 8984  043372  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
 8985  043374  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
 8986  043376  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
 8987  043402  104406                          RESREG                  ;;RESTORE ALL REGISTERS
 8988  043404  000207                          RTS     PC              ;;RETURN TO USER
 8989  043406  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
 8990  043410  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
 8991  043412  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
 8992  043414  006001                          ROR     R1
 8993  043416  006000                          ROR     R0
 8994  043420  006001                          ROR     R1
 8995  043422  006000                          ROR     R0
 8996  043424  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
 8997  043426  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
 8998  043432  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
 8999  043434  000016                  $OCTVL: .BLKB   14.              ;;RESERVE DATA TABLE
 9000
 9001                                  ;THIS ROUTINE IS CALLED BY UNEXPECTED TRAPS TO VECTOR ERRVEC.
 9002                                  ;THE ERROR IS REPORTED AND CONTROL IS TRANSFERRED BACK TO THE TEST
 9003                                  ;FOLLOWING THE ONE THAT WAS INTERRUPTED WHEN THE ERROR OCCURRED!
 9004  043452  011637  001634          CPSPUR: MOV     (SP),$TMP1
 9005  043456  012737  043474  001636          MOV     #1$,$TMP2
 9006  043464  013737  177766  001640          MOV     @#CPUERR,$TMP3
 9007  043472  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
 9008  043474  104150                  1$:     ERROR   150
 9009  043476  104410                          SKIPT
 9010
 9011                                  ;THIS ROUTINE HANDLE UNEXPECTED TRAPS TO #CACHVEC.
 9012  043500  012737  043576  000114  $PUR:   MOV     #10$,@#CACHVEC
 9013  043506  013700  177744                  MOV     @#MEMERR,R0
 9014  043512  032700  000014                  BIT     #14,R0          ;SEE IF IT WAS A MAIN MEMORY ERROR.
 9015  043516  001405                          BEQ     9$
 9016  043520  013701  177740                  MOV     @#LOADRS,R1     ;IF SO THERE IS BAD PARITY IN THE
 9017  043524  042701  176000                  BIC     #176000,R1      ;CACHE AND IT MUST BE PURGED!!!
 9018  043530  005711                          TST     (R1)
 9019  043532  012737  043500  000114  9$:     MOV     #SPUR,@#CACHVEC
 9020  043540  013737  177744  001642          MOV     @#MEMERR,$TMP4  ;TRAP HERE IF AN UNEXPECTED
 9021  043546  013737  177740  001634          MOV     @#LOADRS,$TMP1  ;ERROR, PARITY, OCCURS.
 9022  043554  013737  177742  001636          MOV     @#HIADRS,$TMP2
 9023  043562  011637  001640                  MOV     (SP),$TMP3
 9024  043566  022626                          CMP     (SP)+,(SP)+
 9025  043570  104014                  1$:     ERROR   14
 9026  043572  000005                          RESET                   ;TO STOP THE ACTION OF ANY I/O DEVICE!!!!
 9027  043574  104410                          SKIPT                   ;?????
 9028  043576  022626                  10$:    CMP     (SP)+,(SP)+
 9029  043600  000137  043532                  JMP     9$
 9030
 9031                                  ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL SKIPT.
 9032                                  ;IT TELLS THE USER THAT THE CURRENT TEST HAS BEEN
 9033                                  ;ABORTED AND THAT CONTROL IS BEING PASSED TO THE NEXT TEST.
 9034  043604  011637  001634          ABORTT: MOV     (SP),$TMP1
 9035  043610  112737  000015  001514          MOVB    #15,$ITEMB
 9036  043616  022626                          CMP     (SP)+,(SP)+
 9037  043620  004737  044334                  JSR     PC,ERTYPE
 9038  043624  104407                          RSET
```

```
 9039  043626  000177  000000                    JMP     @SKAD           ;GO TO @SKAD, WHICH SHOULD
 9040                                                                    ;BE SET TO THE
 9041  043632  000000                  SKAD:     .WORD   0               ;ADDRESS OF THE NEXT TEST.
 9042
 9043
 9044                                  ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL RSET. IT CLEARS ALL
 9045                                  ;THE IMPORTANE REGISTERS AND RESETS THE STACK.
 9046  043634                          CLEAN:
 9047
 9048  043634  012737  043500  000114          MOV     #SPUR,@#CACHVEC
 9049  043642  012737  043452  000004          MOV     #CPSPUR,@#ERRVEC
 9050  043650  011637  043722                  MOV     (SP),BACKAD
 9051  043654  012706  001500                  MOV     #STACK,SP
 9052  043660  005037  177750                  CLR     @#MAINT         ;CLEAR ALL CONTROL AND ERROR
 9053  043664  005037  177572                  CLR     @#MMR0          ;REGISTERS.
 9054  043670  005037  172516                  CLR     @#MMR3
 9055  043674  005037  177746                  CLR     @#CONTRL
 9056  043700  012737  177777  177744          MOV     #-1,@#MEMERR
 9057  043706  005037  177766                  CLR     @#CPUERR
 9058  043712  005037  177776                  CLR     @#PSW
 9059  043716  000177  000000                  JMP     @BACKAD
 9060  043722  000000                  BACKAD:   .WORD   0
 9061
 9062
 9063                                  ;COME HERE TO TEST THE REGISTER FLAGS AND USE THEM TO DETERMINE WHETHER
 9064                                  ;OR NOT TO SKIP A TEST WHICH RELIES ON THE FUNCTIONALLITY OF THAT REGISTER
 9065                                  ;TO BE PROPERLY RUN.
 9066                                  ;THESE ROUTINES ARE CALLED BY THE TRAP CATCHER CALLS:
 9067                                  ;       SKPBAD          SKIPT IF BAD ERROR ADDRESS REGISTER
 9068                                  ;       SKPBER          SKIPT IF BAD ERROR REGISTER
 9069                                  ;       SKPBCN          SKIPT IF BAD CONTROL REGISTER
 9070                                  ;       SKPBMN          SKIPT IF BAD MAINTENANCE REGISTER
 9071                                  ;       SKPBHM          SKIPT IF BAD HIT/MISS REGISTER
 9072                                  ;
 9073
 9074  043724  005737  044042  SKBADR: TST     LOAFLG
 9075  043730  001004                  BNE     1$
 9076  043732  005737  044044          TST     HIAFLG
 9077  043736  001001                  BNE     1$
 9078  043740  000002                  RTI
 9079  043742  104400          1$:     TYPE
 9080  043744  053375                  .WORD   ADRNG
 9081  043746  000433                  BR      SKRNG
 9082
 9083  043750  005737  044046  SKBERR: TST     MMRFLG
 9084  043754  001001                  BNE     1$
 9085  043756  000002                  RTI
 9086  043760  104400          1$:     TYPE
 9087  043762  053505                  .WORD   ERRNG
 9088  043764  000424                  BR      SKRNG
 9089
 9090  043766  005737  044050  SKBCNR: TST     CONFLG
 9091  043772  001001                  BNE     1$
 9092  043774  000002                  RTI
 9093  043776  104400          1$:     TYPE
 9094  044000  053605                  .WORD   CNRNG
```

```
9095  044002  000415                        BR      SKRNG
9096
9097  044004  005737  044052        SKBMNR: TST     MANFLG
9098  044010  001001                        BNE     1$
9099  044012  000002                        RTI
9100  044014  104400                1$:     TYPE
9101  044016  053707                        .WORD   MNRNG
9102  044020  000406                        BR      SKRNG
9103
9104  044022  005737  044054        SKBHMR: TST     HIMFLG
9105  044026  001001                        BNE     1$
9106  044030  000002                        RTI
9107  044032  104400                1$:     TYPE
9108  044034  054015                        .WORD   HMRNG
9109
9110  044036  022626                SKRNG:  CMP     (SP)+,(SP)+             ;RESET THE STACK AND GO TO THE
9111  044040  104410                        SKIPT                           ;NEXT TEST!!!!!
9112
9113  044042  000000                LOAFLG: .WORD   0                       ;THESE ARE FLAGS USED TO DESIGNATE
9114  044044  000000                HIAFLG: .WORD   0                       ;EITHER A GOOD OR A BAD REGISTER.
9115  044046  000000                MMRFLG: .WORD   0                       ;GOOD WILL BE DESIGNATED BY A
9116  044050  000000                CONFLG: .WORD   0                       ;0 BAD BY A NOT ZERO!!
9117  044052  000000                MANFLG: .WORD   0
9118  044054  000000                HIMFLG: .WORD   0
9119  044056  000000                LOAFL2: .WORD   0
9120  044060  000000                HIAFL2: .WORD   0
9121  044062  000000                MMRFL2: .WORD   0
9122  044064  000000                CONFL2: .WORD   0
9123  044066  000000                MANFL2: .WORD   0
9124  044070  000000                HIMFL2: .WORD   0
9125
9126                                ;THIS ROUTINE IS CALLED TO DETERMINE THE PARITY OF
9127                                ;A DATA PATTERN. THE PATTERN WHICH IS TAKEN BY THIS
9128                                ;ROUTINE AS ITS ARGUMENT SHOULD BE PUT IN R0. THEN
9129                                ;TRANSFER CONTROL HERE BY EXECUTING:
9130                                ;       JSR     PC,PARCNT
9131                                ;WHEN THIS ROUTINE RETURNS THE NUMBER OF ON,(1), BITS
9132                                ;IN R0 IS LEFT IN R2. THIS WOULD BE A NUMBER BETWEEN
9133                                ;0 AND 16.
9134  044072  012701  000001        PARCNT: MOV     #1,R1
9135  044076  005002                        CLR     R2
9136  044100  030100                1$:     BIT     R1,R0
9137  044102  001401                        BEQ     2$
9138  044104  005202                        INC     R2
9139  044106  006301                2$:     ASL     R1
9140  044110  103373                        BCC     1$
9141  044112  000207                        RTS     PC
9142
9143                                ;THIS ROUTINE IS CALLED TO RESTORE THE TOP 1500 (DEC) WORDS IN THE
9144                                ;FIRST 28K OF MEMORY. THIS SHOULD EFFECTIVELY RESTORE ANY MONITOR
9145                                ;OR LOADER THAT WAS PRESENT BEFORE THIS PROGRAM EEGAN EXECUTION.
9146                                ;CONTROL IS PASSED TO THIS ROUTINE BY AN INTERRUPT FROM THE TTY KEYBOARD
9147                                ;WHEN ANY CHARACTER IS TYPED ON THE KEYBOARD. IF THE CHARACTER.
9148                                ;TURNS OUT TO BE A ↑C (CONTROL-C) THEN MEMORY IS RESTORED. IF THE
9149                                ;CHARACTER IS NOT ↑C THEN A RETURN IS MADE TO THE TEST FOLLOWING
9150                                ;THE ONE WHOSE EXECUTION WAS INTERRUPTED BY THE KEYBOARD INTERRUPT.
```

```
 9151  044114  017700  135426          RESMON: MOV     @$TKB,R0
 9152  044120  104407                          RSET
 9153  044122  005003                          CLR     R3
 9154  044124  042700  000200                  BIC     #BIT7,R0        ;GET THE CHARACTER, INITIALIZE THE REGISTERS
 9155  044130  022700  000003                  CMP     #3,R0           ;AND SEE IF THE CHARACTER WAS ↑C.
 9156  044134  001027                          BNE     NOCNC           ;BRANCH AND GO TO NEXT TEST IF NOT.
 9157  044136  104400                          TYPE                    ;ECHOE THE CONTROL-C AS '↑C'
 9158  044140  052350                          .WORD   CONCMS
 9159  044142  012704  002734          CHAINQ: MOV     #↑D1500,R4              ;AND RESTORE THE MONITOR.
 9160  044146  012701  073446                  MOV     #BOTTOM+4,R1
 9161  044152  012702  160000                  MOV     #160000,R2
 9162  044156  012142                  1$:     MOV     (R1)+,-(R2)
 9163  044160  077402                          SOB     R4,1$
 9164  044162  012737  177777  044234          MOV     #-1,MONF        ;RESET THE MONITOR RESTORED FLAG.
 9165  044170  022703  125252                  CMP     #125252,R3
 9166  044174  001001                          BNE     STOP
 9167  044176  000207                          RTS     PC              ;IF THE MONITOR WAS RESTORED BY THE
 9168                                                                  ;.$EOP ROUTIN RETURN TO .$EOP.
 9169                                                                  ;OTHERWISE HALT.
 9170  044200  104400          STOP:   TYPE                            ;TYPE THE MONITOR RESTORED MESSAGE.
 9171  044202  052354                  .WORD   MMESRS
 9172  044204  013737  044232  000060          MOV     MONTTY,@#TKVEC  ;SET THE TTY KEYBOARD INTERRUPT VECTOR
 9173                                                                  ;TO ITS INITIAL STATE.
 9174  044212  000000                          HALT                    ;AND HALT!!
 9175  044214  005077  135326          NOCNC:  CLR     @$TKB           ;NOT CONTROL C SO RETURN TO NEXT TEST.
 9176  044220  152777  000100  135316          BISB    #BIT6,@$TKS
 9177  044226  000177  177400                  JMP     @SKAD           ;RETURN.
 9178  044232  000000          MONTTY: .WORD   0       ;STORAGE FOR THE TTY KEYBOARD VECTOR'S ORIGINAL
 9179                                                                  ;CONTENTS.
 9180  044234  177777          MONF:   .WORD   177777                  ;FLAG, IF NOT -1 THE MONITOR IS SAVED!!
 9181
 9182
 9183                                  ;THIS ROUTINE IS CALLED BY THE TRAP CALL MMSKIP. IT LOOKS
 9184                                  ;AT THE SWITCH REGISTER AND DETERMINES WHETHER OR NOT
 9185                                  ;SWITCH #7 IS ON. IF SO THE CURRENT TEST IS SKIPPED
 9186                                  ;AND THE NEXT TEST IS ENTERED. A SSKAD MUST BE ISSUED
 9187                                  ;BEFORE THE MMSKIP.
 9188                                  ;THE PURPOSE OF SWITCH #7 IS TO CAUSE THE DELETION OF THE
 9189                                  ;EXECUTION OF ANY TEST WHICH RELIES ON MEMORY MANAGEMENT
 9190                                  ;FOR ITS OPERATION.
 9191
 9192  044236  032777  000200  135274  MMDES:  BIT     #SW7,@SWR
 9193  044244  001001                          BNE     1$              ;IS THE SWITCH ON?
 9194  044246  000002                          RTI                     ;NO, SO RETURN.
 9195  044250  022626                  1$:     CMP     (SP)+,(SP)+
 9196  044252  104407                          RSET
 9197  044254  000177  177352                  JMP     @SKAD           ;YES, GO TO THE NEXT TEST.
 9198                                  ;THIS ROUTINE IS CALLED TO DETERMINE THE HIGHEST POSSIBLE
 9199                                  ;ADDRESS IN MEMORY. IT IS CALLED THUS, BY TRAP CALL SIZE:
 9200                                  ;       SIZE
 9201                                  ;LOORDA: .WORD  0
 9202                                  ;HIORDA: .WORD  0
 9203                                  ;NXTINST:
 9204                                  ;THE LOW ORDER 16-BITS OF THE ADDRESS ARE LEFT IN THE
 9205                                  ;WORD DIRECTLY FOLLOWING THE CALL. THE HIGH ORDER 6-BITS
 9206                                  ;ARE LEFT IN THE NEXT WORD AND CONTROL IS RETURNED
```

```
                                        ;TO THE THIRD WORD FOLLOWING THE CALL.
044260  010046                  MSIZER: MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0 AND R1
044262  010146                          MOV     R1,-(SP)        ;GET THE ADDRESS OF
044264  016600  000004                  MOV     4(SP),R0        ;THE CALL OF THE STACK.
044270  013710  177760                  MOV     @#SIZELO,(R0)
044274  005060  000002                  CLR     2(R0)
044300  012701  000006                  MOV     #6,R1           ;ROTATE THE 16-BIT 'BLOCK'
                                                                ;NUMBER 6-BITS TO THE
044304  006310                  1$:     ASL     (R0)            ;LEFT AND TURN ON LOW ORDER
044306  006160  000002                  ROL     2(R0)           ;BITS 1-5 LEAVING BIT-0
044312  077104                          SOB     R1,1$           ;OFF SO AS TO CREATE
044314  052710  000076                  BIS     #76,(R0)        ;THE 22-BIT PHYSICAL ADDRESS OF
                                                                ;THE HIGHEST WORD IN
                                                                ;MEMORY.
044320  022020                          CMP     (R0)+,(R0)+     ;DETERMINE THE RETURN ADDRESS
044322  010066  000004                  MOV     R0,4(SP)        ;AND LEAVE ON THE STACK FOR
                                                                ;AN RTI.
044326  012601                          MOV     (SP)+,R1        ;RESTORE R1 AND R0.
044330  012600                          MOV     (SP)+,R0
044332  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
044334  104400                  ERTYPE: TYPE
044336  001713                          .WORD   $CRLF
044340  010046                          MOV     R0,-(SP)        ;SAVE R0
044342  005000                          CLR     R0
044344  113700  001514                  MOVB    $ITEMB,R0       ;GET THE ITEM NUMBER
044350  001005                          BNE     1$              ;ZERO?
044352  013746  001516                  MOV     $ERRPC,-(SP)    ;YES, TYPE JUST THE PC
044356  104401                          TYPOC                   ;OF THE ERROR CALL.
044360  000137  044676                  JMP     ERT5

044364  005300                  1$:     DEC     R0              ;MAKE R0 AN INDEX FOR THE
044366  072027  000003                  ASH     #3,R0           ;ERROR TABLE
044372  062700  001716                  ADD     #$ERRTB,R0
044376  012037  044406                  MOV     (R0)+,2$        ;TYPE EM, ERROR MESSAGE.
044402  001404                          BEQ     3$
044404  104400                          TYPE
044406  000000                  2$:     .WORD   0
044410  104400                          TYPE
044412  001713                          .WORD   $CRLF
044414  012037  044424          3$:     MOV     (R0)+,4$        ;TYPE DH, DATA HEADER
044420  001404                          BEQ     5$
044422  104400                          TYPE
044424  000000                  4$:     .WORD   0
044426  104400                          TYPE
044430  001713                          .WORD   $CRLF
044432  010146                  5$:     MOV     R1,-(SP)        ;SAVE R1
044434  012001                          MOV     (R0)+,R1        ;GET DT, DATA TABLE ADDRESS
044436  001002                          BNE     6$
044440  000137  044674                  JMP     ERT4            ;JMP IF NO ERROR TABLE.
044444  012000                  6$:     MOV     (R0)+,R0        ;GET DF, DATA FORMAT ADDRESS
044446  105710                  ERT1:   TSTB    (R0)            ;DATA FORMAT ENTRY EQUALS
```

```
9262  044450  001003                          BNE     7$              ;ZERO?
9264  044452  013146                          MOV     @(R1)+,-(SP)    ;YES, SO TYPE A 16-BIT
9265  044454  104401                          TYPOC                   ;OCTAL NUMBER
9266  044456  000500                          BR      ERT2
9267  044460  122710  000001          7$:     CMPB    #1,(R0)         ;FORMAT EQUALS 1?
9268  044464  001003                          BNE     8$
9269  044466  013146                          MOV     @(R1)+,-(SP)    ;YES, TYPE A DECIMAL NUMBER
9270  044470  104404                          TYPDS
9271  044472  000472                          BR      ERT2
9272
9273  044474  122710  000002          8$:     CMPB    #2,(R0)         ;FORMAT 2?
9274  044500  001012                          BNE     9$
9275  044502  013146                  85$:    MOV     (R1)+,-(SP)     ;YES, TYPE A 22-BIT NUMBR
9276  044504  004737  043332                  JSR     PC,$DB2O        ;CALL $DB2O TO CONVERT THE
9277  044510  062716  000003                  ADD     #3,(SP)         ;BINARY TO ASCII
9278  044514  012637  044522                  MOV     (SP)+,29$       ;TYPE THE STRING
9279  044520  104400                          TYPE
9280  044522  000000                  29$:    .WORD   0
9281  044524  000455                          BR      ERT2
9282
9283  044526  122710  000004          9$:     CMPB    #4,(R0)         ;FORMAT 4?
9284  044532  001004                          BNE     10$
9285  044534  013146                          MOV     @(R1)+,-(SP)    ;YES, TYPE A 16-BIT
9286  044536  104402                          TYPOS                   ;OCTAL NUMBER SUPRESSING
9287  044540     016                          .BYTE   16              ;LEADING ZEROES
9288  044541     000                          .BYTE   0
9289  044542  000446                          BR      ERT2
9290  044544  122710  000003          10$:    CMPB    #3,(R0)         ;FORMAT 3?
9291  044550  001007                          BNE     11$
9292  044552  013146                          MOV     @(R1)+,-(SP)    ;YES CONVERT 16-BIT
9293  044554  012737  177777  044702          MOV     #-1,TVADFL      ;VIRTUAL ADDRESS TO 32-BIT
9294  044562  004737  044710                  JSR     PC,TYPVAD       ;PHYSICAL ADDRESS AND TYPE
9295  044566  000434                          BR      ERT2            ;RELOCATE ONLY IF SEG. IS ON!
9296  044570  122710  000005          11$:    CMPB    #5,(R0)         ;FORMAT 5?
9297  044574  001005                          BNE     12$
9298  044576  012137  044604                  MOV     (R1)+,20$       ;PRINT ASCIZ STRING
9299  044602  104400                          TYPE
9300  044604  000000                  20$:    .WORD   0
9301  044606  000426                          BR      ERT3
9302
9303  044610  122710  000006          12$:    CMPB    #6,(R0)         ;FORMAT 6
9304  044614  001005                          BNE     13$
9305  044616  005037  044702                  CLR     TVADFL
9306  044622  004737  044710                  JSR     PC,TYPVAD
9307  044626  000414                          BR      ERT2
9308
9309  044630  122710  000007          13$:    CMPB    #7,(R0)         ;FORMAT 7?
9310  044634  001010                          BNE     14$
9311  044636  012146                          MOV     (R1)+,-(SP)
9312  044640  004737  043332                  JSR     PC,$DB2O
9313  044644  012637  044652                  MOV     (SP)+,45$
9314  044650  104400                          TYPE
9315  044652  000000                  45$:    .WORD   0
9316  044654  000401                          BR      ERT2
9317
9318  044656  000000          14$:    HALT                            ;?????
```

```
 9319
 9320  044660  104400                ERT2:   TYPE                    ;PRINT A TAB AFTER TYPING AN
 9321  044662  052460                        .WORD   $TAB            ;ERROR TABLE ENTRY OF ALL MODES
 9322                                                                ;EXCEPT ASCIZ
 9323  044664  005200                ERT3:   INC     R0              ;POINT TO THE NEXT FORMAT BYTE
 9324  044666  005711                        TST     (R1)            ;IS THERE ANOTHER ENTRY?
 9325  044670  001401                        BEQ     ERT4
 9326  044672  000665                        BR      ERT1            ;YES, PROCESS IT
 9327                                                                ;OTHERWISE:
 9328  044674  012601                ERT4:   MOV     (SP)+,R1        ;RESTORE R1
 9329  044676  012600                ERT5:   MOV     (SP)+,R0        ;RESTORE R0
 9330  044700  000207                        RTS     PC              ;AND RETURN
 9331
 9332  044702  000000                TVADFL: .WORD   0               ;FLAG USED TO TELL TYVAD
 9333                                                                ;WHETHER TO CONDITIONALLY
 9334                                                                ;OR UNCONDITIONALLY RELOCATE
 9335                                                                ;WHEN TYPING AN ADDRESS,
 9336                                                                ;-1 OR 0 RESPECTIVELY
 9337
 9338  044704  000000                TVADLO: .WORD   0               ;REGISTERS FOR THE 22-BIT
 9339  044706  000000                TVADHI: .WORD   0               ;ADDRESS COMPUTED BY TYVAD.
 9340
 9341                                ;ROUTINE WHICH CONVERTS A 16-BIT ADDRESS TO A 22-BIT
 9342                                ;ADDRESS. IF TVADFL IS -1, THEN CONVERT TO THE 22-BIT
 9343                                ;REAL ADDRESS DEPENDENT ON SEG BEING ON OR OFF FOR RELOCATION.
 9344                                ;IF TVADFL IS ZERO THEN UNCONDITIONAL USE THE KERNAL
 9345                                ;PAR WHICH IS APPROPIATE TO DO RELOCATION.
 9346  044710  104405                TYPVAD: SAVREG
 9347  044712  016601  000002               MOV     2(SP),R1        ;GET THE VIRTUAL
 9348  044716  010137  044704               MOV     R1,TVADLO       ;ADDRESS
 9349  044722  005037  044706               CLR     TVADHI
 9350  044726  005737  044702               TST     TVADFL          ;CONDITIONALLY RELOCATE?
 9351  044732  001404                       BEQ     1$
 9352  044734  032737  000001  177572       BIT     #1,@#MMR0       ;YES, SEE IF MEMORY
 9353  044742  001424                       BEQ     2$              ;MANAGEMENT IS ON
 9354  044744  005000                1$:    CLR     R0              ;RELOCATE
 9355  044746  073027  000003               ASHC    #3,R0           ;LEFT SHIFT R0 AND R1
 9356  044752  006300                       ASL     R0              ;THREE PLACES. R0 ONE
 9357                                                                ;MORE SO THAT IT CONTAINS
 9358                                                                ;2 X THE UPPER 3-BITS OF
 9359  044754  000241                       CLC                     ;THE VIRTUAL ADDRESS
 9360  044756  006001                       ROR     R1              ;RESTORE R1 TO THE OFFSET
 9361  044760  006001                       ROR     R1              ;OF THE VIRTUAL ADDRESS
 9362  044762  006001                       ROR     R1              ;TO THE PAR
 9363  044764  062700  172340               ADD     #KIPAR0,R0      ;DETERMINE THE CORRECT PAR'S
 9364                                                                ;ADDRESS
 9365  044770  011003                       MOV     (R0),R3         ;GET ITS CONTENTS
 9366  044772  005002                       CLR     R2
 9367  044774  073227  000006               ASHC    #6,R2           ;MAKE THE BLOCK COUNT
 9368                                                                ;A 22-BIT ADDRESS.
 9369  045000  060103                       ADD     R1,R3           ;ADD THE OFFSET TO THE
 9370  045002  005502                       ADC     R2              ;BASE ADDRESS
 9371
 9372  045004  010237  044706               MOV     R2,TVADHI
 9373  045010  010337  044704               MOV     R3,TVADLO
 9374  045014  012746  044704        2$:    MOV     #TVADLO,-(SP)   ;CALL $DB2O TO CONVERT THE
```

```
 9375  045020  004737  043332                JSR     PC,$DB2O        ;22-BIT
 9376  045024  062716  000003                ADD     #3,(SP)         ;TYPE ONLY 8 DIGITS.
 9377  045030  012637  045036                MOV     (SP)+,3$
 9378  045034  104400                        TYPE
 9379  045036  000000              3$:       .WORD   0
 9380  045040  104406                        RESREG                  ;RESTORE THE REGISTERS
 9381  045042  012616                        MOV     (SP)+,(SP)      ;LEAVE ONLY THE RETURN
 9382                                                                ;ADDRESS ON THE STACK.
 9383  045044  000207                        RTS     PC              ;RETURN
 9384
 9385                                        .SBTTL  SYSTEM DEVICE SIZER
 9386
 9387                                ;THIS ROUTINE IS CALLED TO DETERMINE WHAT
 9388                                ;CONTROLLERS AND WHAT DRIVES ARE AVAILABLE ON
 9389                                ;THE SYSTEM.
 9390                                ;IT USES THE FLAGS:
 9391                                ;               RS4DFL
 9392                                ;               RP4DFL
 9393                                ;               RH4DFL
 9394                                ;               RK5DFL
 9395                                ;               UBEDFL
 9396                                ;WHICH ARE BYTES CONTAINING A BIT FOR EACH
 9397                                ;POSSIBLE DEVICE ON THE CONTROLLER
 9398  045046  005037  046754        SIZDEV: CLR     RS4FLG
 9399  045052  005037  046012                CLR     RP4FLG
 9400  045056  005037  051456                CLR     RH4FLG
 9401  045062  005037  047710                CLR     RK5FLG
 9402  045066  005037  050724                CLR     UBEFLG
 9403  045072  005037  046756                CLR     RS4ER1
 9404  045076  005037  046014                CLR     RP4ER1
 9405  045102  005037  051460                CLR     RH4ER1
 9406  045106  005037  047712                CLR     RK5ER1
 9407  045112  005037  050726                CLR     UBEER1
 9408  045116  104405                        SAVREG
 9409  045120  105037  045440                CLRB    RS4DFL
 9410  045124  105037  045441                CLRB    RP4DFL
 9411  045130  105037  045442                CLRB    RH4DFL
 9412  045134  105037  045443                CLRB    RK5DFL
 9413  045140  105037  045444                CLRB    UBEDFL
 9414
 9415  045144  013737  000004  045446        MOV     @#4,SIZTM1
 9416  045152  012737  045200  000004        MOV     #1$,@#4
 9417
 9418  045160  005777  136344                TST     @RS4CS1
 9419  045164  004737  045460                JSR     PC,SETREG
 9420  045170  003526                        .WORD   RS4REG
 9421  045172  004737  045520                JSR     PC,SIZRS4
 9422  045176  000403                        BR      2$
 9423
 9424  045200  022626              1$:       CMP     (SP)+,(SP)+
 9425  045202  005037  177766                CLR     @#CPUERR
 9426
 9427  045206  012737  045234  000004  2$:   MOV     #3$,@#4
 9428
 9429  045214  005777  136346                TST     @RP4CS1
 9430  045220  004737  045460                JSR     PC,SETREG
```

```
 9431  045224  003564                          .WORD   RP4REG
 9432  045226  004737  045604                  JSR     PC,SIZRP4
 9433  045232  000403                          BR      6$
 9434
 9435  045234  022626                  3$:     CMP     (SP)+,(SP)+
 9436  045236  005037  177766                  CLR     @#CPUERR
 9437
 9438  045242  012737  045330  000004  6$:     MOV     #7$,@#4
 9439  045250  005777  136370                  TST     @RH4CS1
 9440  045254  004737  045460                  JSR     PC,SETREG
 9441  045260  003642                          .WORD   RH4REG
 9442  045262  012777  000007  136364          MOV     #7,@RH4CS2
 9443  045270  022777  000040  136374          CMP     #40,@RH4DT
 9444  045276  001017                          BNE     8$
 9445  045300  013737  003644  003676          MOV     RH4CS1,RH4AE
 9446  045306  062737  000074  003676          ADD     #74,RH4AE
 9447  045314  004737  045460                  JSR     PC,SETREG
 9448  045320  003674                          .WORD   RH4REX
 9449
 9450  045322  004737  046002                  JSR     PC,SIZRH4
 9451  045326  000403                          BR      8$
 9452
 9453  045330  022626                  7$:     CMP     (SP)+,(SP)+
 9454  045332  005037  177766                  CLR     @#CPUERR
 9455
 9456  045336  012737  045364  000004  8$:     MOV     #9$,@#4
 9457  045344  005777  136334                  TST     @RK5DS
 9458  045350  004737  045460                  JSR     PC,SETREG
 9459  045354  003702                          .WORD   RK5REG
 9460  045356  004737  045670                  JSR     PC,SIZRK5
 9461  045362  000403                          BR      10$
 9462
 9463  045364  022626                  9$:     CMP     (SP)+,(SP)+
 9464  045366  005037  177766                  CLR     @#CPUERR
 9465
 9466  045372  012737  045420  000004  10$:    MOV     #11$,@#4
 9467  045400  005777  136320                  TST     @UBEDB
 9468  045404  004737  045460                  JSR     PC,SETREG
 9469  045410  003722                          .WORD   UBEREG
 9470  045412  004737  045754                  JSR     PC,SIZUBE
 9471  045416  000403                          BR      12$
 9472
 9473  045420  022626                  11$:    CMP     (SP)+,(SP)+
 9474  045422  005037  177766                  CLR     @#CPUERR
 9475
 9476  045426  013737  045446  000004  12$:    MOV     SIZTM1,@#4
 9477  045434  104406                          RESREG
 9478  045436  000207                          RTS     PC
 9479
 9480  045440     000                  RS4DFL: .BYTE   0
 9481  045441     000                  RP4DFL: .BYTE   0
 9482  045442     000                  RH4DFL: .BYTE   0
 9483  045443     000                  RK5DFL: .BYTE   0
 9484  045444     000                  UBEDFL: .BYTE   0
 9485          045446                          .EVEN
 9486
```

```
 9487  045446  000000                  SIZTM1: .WORD   0
 9488  045450  000000                  SIZTM2: .WORD   0
 9489  045452  000000                  SIZTM3: .WORD   0
 9490  045454  000000                  SIZTM4: .WORD   0
 9491  045456  000000                  SIZTM5: .WORD   0
 9492
 9493                                  ;THIS ROUTINE IS CALED BY A:
 9494                                  ;               JSR     PC,SETREG
 9495                                  ;               .WORD   DEVREG
 9496                                  ;WHERE DEVREG IS THE STARTING ADDRESS OF
 9497                                  ;A TABLE, WHICH IS TO CONTAIN THE ADDRESS OF
 9498                                  ;A DEVICE'S CONTROL AND STATUS REGISTERS.
 9499                                  ;THE TABLES ARE GENERATE HERE
 9500
 9501  045460  011637  045516          SETREG: MOV     (SP),SETMP
 9502  045464  062716  000002                  ADD     #2,(SP)
 9503  045470  104405                          SAVREG
 9504  045472  017700  000020                  MOV     @SETMP,R0
 9505  045476  012001                          MOV     (R0)+,R1
 9506  045500  011002                          MOV     (R0),R2
 9507  045502  010220                  1$:     MOV     R2,(R0)+
 9508  045504  062702  000002                  ADD     #2,R2
 9509  045510  077104                          SOB     R1,1$
 9510  045512  104406                          RESREG
 9511  045514  000207                          RTS     PC
 9512
 9513  045516  000000                  SETMP:  .WORD   0
 9514
 9515
 9516                                  ;THIS ROUTINE IS CALLED, AFTER IT HAS BEEN
 9517                                  ;DETERMINED IF THERE IS A RS04 CONTROLLER, TO SEE
 9518                                  ;WHT DRIVES ARE AVAILABLE.
 9519
 9520  045520  012700  000010          SIZRS4: MOV     #10,R0
 9521  045524  012701  000001                  MOV     #1,R1
 9522  045530  005002                          CLR     R2
 9523  045532  105037  045541                  CLRB    3$
 9524
 9525  045536  104421                  1$:     CALRS4
 9526  045540     001                  2$:     .BYTE   1
 9527  045541     000                  3$:     .BYTE   0               ;DO A NOP FUNCTION
 9528  045542  000000                          .WORD   0               ;FOR EACH OF POSSIBLY
 9529  045544  000000                          .WORD   0               ;8 DRIVES
 9530  045546  000000                          .WORD   0
 9531  045550  000000                          .WORD   0
 9532  045552  000000                          .WORD   0
 9533  045554  000000                          .WORD   0
 9534
 9535  045556  005737  046756                  TST     RS4ER1
 9536  045562  001001                          BNE     4$
 9537  045564  050102                          BIS     R1,R2
 9538  045566  006301                  4$:     ASL     R1
 9539  045570  105237  045541                  INCB    3$
 9540  045574  077020                          SOB     R0,1$
 9541
 9542  045576  110237  045440                  MOVB    R2,RS4DFL
```

```
 9543  045602  000207                          RTS     PC
 9544
 9545                                  ;THIS ROUTINE IS CALLED TO DETERMINE WHAT RP04
 9546                                  ;DRIVES ARE ON THE CONTROLLER
 9547
 9548  045604  012700  000010          SIZRP4: MOV     #10,R0
 9549  045610  012701  000001                  MOV     #1,R1
 9550  045614  005002                          CLR     R2
 9551  045616  105037  045625                  CLRB    3$
 9552
 9553  045622  104422                  1$:     CALRP4                  ;DO A READ IN PRESET
 9554  045624     021                  2$:     .BYTE   21              ;FOR EACH OF UP TO
 9555  045625     000                  3$:     .BYTE   0               ;8 DRIVES.
 9556  045626  000000                          .WORD   0
 9557  045630  000000                          .WORD   0
 9558  045632  000000                          .WORD   0
 9559  045634  000000                          .WORD   0
 9560  045636  000000                          .WORD   0
 9561  045640  000000                          .WORD   0
 9562
 9563  045642  005737  046014                  TST     RP4ER1
 9564  045646  001001                          BNE     4$
 9565  045650  050102                          BIS     R1,R2
 9566  045652  006301                  4$:     ASL     R1
 9567  045654  105237  045625                  INCB    3$
 9568  045660  077020                          SOB     R0,1$
 9569
 9570  045662  110237  045441                  MOVB    R2,RP4DFL
 9571  045666  000207                          RTS     PC
 9572
 9573                                  ;DETERMINE WHAT RK05 DRIVES ARE AVAILABLE.
 9574
 9575  045670  012700  000010          SIZRK5: MOV     #10,R0
 9576  045674  012701  000001                  MOV     #1,R1
 9577  045700  005002                          CLR     R2
 9578  045702  105037  045711                  CLRB    3$
 9579
 9580  045706  104424                  1$:     CALRK5                  ;DO A DRIVE RESET
 9581  045710     015                  2$:     .BYTE   15              ;FOR EACH OF 8
 9582  045711     000                  3$:     .BYTE   0               ;POSSIBLE DRIVES.
 9583  045712  000000                          .WORD   0
 9584  045714  000000                          .WORD   0
 9585  045716  000000                          .WORD   0
 9586  045720  000000                          .WORD   0
 9587  045722  000000                          .WORD   0
 9588  045724  000000                          .WORD   0
 9589
 9590  045726  005737  047712                  TST     RK5ER1
 9591  045732  001001                          BNE     4$
 9592  045734  050102                          BIS     R1,R2
 9593  045736  006301                  4$:     ASL     R1
 9594  045740  105237  045711                  INCB    3$
 9595  045744  077020                          SOB     R0,1$
 9596
 9597  045746  110237  045443                  MOVB    R2,RK5DFL
 9598  045752  000207                          RTS     PC
```

```
 9599
 9600                                   ;SET UP UBEDFL
 9601
 9602  045754  042777  000200  135750  SIZUBE: BIC     #BIT7,@UBECR1
 9603  045762  032777  000200  135742          BIT     #BIT7,@UBECR1
 9604  045770  001403                          BEQ     1$
 9605  045772  112737  000001  045444          MOVB    #1,UBEDFL
 9606  046000  000207                  1$:     RTS     PC
 9607
 9608                                   ;DETERMINE WHAT MASS BUS TESTER UNITS THERE ARE
 9609
 9610  046002  012737  000200  045442  SIZRH4: MOV     #BIT7,RH4DFL
 9611  046010  000207                          RTS     PC
 9612
 9613                                           .SBTTL  DEVICE HANDLERS
 9614                                   ;;*******************************************************************
 9615                                   ;*
 9616                                   ;*      THE FOLLOWING SIX ROUTINES:
 9617                                   ;*              RH4HAN
 9618                                   ;*              RP4HAN
 9619                                   ;*              RS4HAN
 9620                                   ;*              UBEHAN
 9621                                   ;*              RK5HAN
 9622                                   ;*      ARE O/I AND BUS TESTER DEVICE HANDLERS.
 9623                                   ;*      THEY ARE CALLED USING:
 9624                                   ;*              TRAP TABLE CALL
 9625                                   ;*      FUNCTION:.BYTE
 9626                                   ;*              UNITNUM:.BYTE
 9627                                   ;*      DISKADR1:.WORD
 9628                                   ;*      DISKADR2:.WORD
 9629                                   ;*      MEMADR1:.WORD
 9630                                   ;*      MEMADR2:.WORD
 9631                                   ;*      WORDCNT:.WORD
 9632                                   ;*      VECTOR: .WORD
 9633                                   ;*      RETURN:
 9634                                   ;*A     WHERE TRAP TABLE CALL IS ONE OF:
 9635                                   ;*              CALRH4
 9636                                   ;*              CALRP4
 9637                                   ;*              CALRS4
 9638                                   ;*              CALUBE
 9639                                   ;*              CALRK5
 9640                                   ;*B     FUNCTION IS THE PATTERN TO BE LOADED INTO THE
 9641                                   ;*      CONTROL REGISTER FUNCTION BITS, WITH EITHER
 9642                                   ;*      INTERRUPT ENABLED OR NOT.
 9643                                   ;*C     UNITNUM IS THE DRIVE NUMBER
 9644                                   ;*D     DISKADR1 AND DISKADR2 ARE THE DISK ADDRESS
 9645                                   ;*      SECTOR NUMBER
 9646                                   ;*E     MEMADR1 AND MEMADR2 ARE THE 22-BIT MEMORY
 9647                                   ;*      ADDRESS FOR THE TRANSFER.
 9648                                   ;*F     WORDCNT IS THE WORD COUNT A POSITIVE
 9649                                   ;*      NUMBER BETWEEN 0 AND 32K.
 9650                                   ;*G     VECTOR IS THE INTERRUPT HANDLER ROUTINE SPECIFIED
 9651                                   ;*      BY THE USER FOR AN INTERRUPT ENABLED FUNCTION.
 9652                                   ;*
 9653                                   ;*      WHEN THE HANDLER PROCESSES A CALL IT RETURNS
 9654                                   ;*      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
046012  000000                          RP4FLG:.WORD    0
046014  000000                          RP4ER1:.WORD    0               ;ERROR FLAGS.
046016  000000                          RP4ER2:.WORD    0
046020  000000                          RP4ER3:.WORD    0
046022  000000                          RP4ER4:.WORD    0
046024  000000                          RP4USE:.WORD    0
046026  000000                          RP4TMP:.WORD    0
046030  000000                          RP4FUN:.WORD    0
046032  000000                          RP4UNI:.WORD    0
046034  000000                          RP4DA1:.WORD    0
046036  000000                          RP4DA2:.WORD    0
046040  000000                          RP4MA1:.WORD    0
046042  000000                          RP4MA2:.WORD    0
046044  000000                          RP4WCT:.WORD    0
046046  000000                          RP4VEC:.WORD    0
046050  000000                          RP4TRK:.WORD    0
046052  000000                          RP4SEC:.WORD    0
046054  000000                          RP4CYL:.WORD    0

046056  005737  046012          RP4HAN: TST     RP4FLG          ;SEE IF THERE IS
046062  001402                          BEQ     RP4H1           ;ALREADY AN RP04 FUNCTION
046064  104000                          ERROR                   ;IN PROGRESS. IF THERE
046066  000000                          HALT                    ;IS ERROR> (SHOULD NEVER
046070  012737  000340  177776  RP4H1:  MOV     #340,@#PSW      ;HAPPEN.)
046076  011637  046026                  MOV     (SP),RP4TMP     ;RAISE THE PRIORITY
046102  062716  000016                  ADD     #16,(SP)
046106  104405                          SAVREG                  ;GET AN ARGUMENT POINTER
046110  013700  046026                  MOV     RP4TMP,R0       ;RESET THE RETURN ADDRESS
046114  112037  046030                  MOVB    (R0)+,RP4FUN    ;FUNCTION
046120  112037  046032                  MOVB    (R0)+,RP4UNI    ;UNIT, DEVICE, NUMBER
046124  012037  046034                  MOV     (R0)+,RP4DA1    ;DISK ADDRESS
046130  012037  046036                  MOV     (R0)+,RP4DA2
046134  012037  046040                  MOV     (R0)+,RP4MA1    ;MEMORY ADDRESS
```

```
9711  046140  012037  046042                  MOV     (R0)+,RP4MA2
9712  046144  012037  046044                  MOV     (R0)+,RP4WCT    ;WORD COUNT
9713  046150  012037  046046                  MOV     (R0)+,RP4VEC    ;INTERRUPT HANDLER ROUTINE
9714  046154  005037  046014                  CLR     RP4ER1          ;CLEAR THE ERROR
9715  046160  005037  046016                  CLR     RP4ER2          ;FLAGS
9716  046164  005037  046020                  CLR     RP4ER3
9717
9718  046170  004737  046452                  JSR     PC,RP4S1        ;GO SET UP THE UNIT NUMBER
9719  046174  004737  046522                  JSR     PC,RP4RDY       ;GET THE DEVICE READY.
9720  046200  004737  046462                  JSR     PC,RP4S2        ;COMPUTE THE CYLINDER,
9721                                                                  ;TRACK AND SECTOR
9722  046204  004737  046506                  JSR     PC,RP4S3        ;SET UP THE WORD COUNT
9723
9724  046210  013777  046032  135360  RP4H2:  MOV     RP4UNI,@RP4CS2  ;SET THE RP04 REGISTERS
9725  046216  013777  046044  135344          MOV     RP4WCT,@RP4WC   ;UP FOR THIS FUNCTION
9726  046224  013777  046040  135340          MOV     RP4MA1,@RP4BA
9727  046232  013777  046042  135376          MOV     RP4MA2,@RP4BAE
9728  046240  013777  046036  135326          MOV     RP4DA2,@RP4DA
9729  046246  013777  046034  135346          MOV     RP4DA1,@RP4DC
9730  046254  013700  003742                  MOV     RP4V,R0         ;SET UP THE INTERRUPT
9731  046260  012720  046332                  MOV     #RP4H4,(R0)+    ;VECTOR
9732  046264  012710  000340                  MOV     #340,(R0)
9733  046270  013700  046030                  MOV     RP4FUN,R0       ;LOAD THE FUNCTION
9734  046274  010037  046012                  MOV     R0,RP4FLG       ;AND GO
9735  046300  110077  135262                  MOVB    R0,@RP4CS1
9736  046304  032700  000100                  BIT     #BIT6,R0        ;SEE IF THE FUNCTION
9737  046310  001402                          BEQ     RP4H3           ;WILL INTERRUPT WHEN
9738  046312  104406                          RESREG                  ;DONE. IF YES RETURN
9739  046314  000002                          RTI                     ;IF NOT INTERRUPTING
9740  046316  004737  046346          RP4H3:  JSR     PC,RP4H5        ;THEN WAIT FOR THE
9741  046322  005037  046012                  CLR     RP4FLG          ;FUNCTION TO FINISH.
9742  046326  104406                          RESREG                  ;THEN RETURN.
9743  046330  000002                          RTI
9744
9745  046332  005037  046012          RP4H4:  CLR     RP4FLG          ;WHEN THE INTERRUPT
9746  046336  004737  046346                  JSR     PC,RP4H5        ;OCCURS CHECK FOR ERRORS
9747  046342  000177  177500                  JMP     @RP4VEC         ;AND GO TO THE SERVICE
9748                                                                  ;ROUTINE.
9749
9750  046346  010046                  RP4H5:  MOV     R0,-(SP)
9751  046350  053777  046032  135220  RP4H51: BIS     RP4UNI,@RP4CS2
9752  046356  017700  135204                  MOV     @RP4CS1,R0
9753  046362  005700                          TST     R0              ;SEE IF THE FUNCTION
9754  046364  100023                          BPL     RP4H6           ;WAS COMPLETED WITHOUT
9755  046366  032700  060000                  BIT     #60000,R0       ;ERRORS.
9756  046372  001420                          BEQ     RP4H6
9757  046374  017737  135176  046016          MOV     @RP4CS2,RP4ER2  ;IF ERRORS OCCURRED SET
9758  046402  017737  135172  046020          MOV     @RP4DS,RP4ER3   ;THE INDICATORS
9759  046410  017737  135166  046022          MOV     @RP4RR1,RP4ER4
9760  046416  012737  177777  046014          MOV     #-1,RP4ER1
9761  046424  004737  046736                  JSR     PC,RP4CLR       ;CLEAR THE CONTROL
9762  046430  012600                          MOV     (SP)+,R0
9763  046432  000207                          RTS     PC
9764  046434  105700                  RP4H6:  TSTB    R0              ;WAIT FOR READY OR
9765  046436  100344                          BPL     RP4H51          ;ERROR
9766  046440  105777  135134                  TSTB    @RP4DS
```

```
 9767  046444  100341                          BPL     RP4HS1
 9768  046446  012600                          MOV     (SP)+,R0
 9769  046450  000207                          RTS     PC
 9770
 9771  046452  042737  177770  046032  RP4S1:  BIC     #177770,RP4UNI  ;SET UP THE DRIVE NUMBER.
 9772  046460  000207                          RTS     PC
 9773
 9774  046462  013701  046034          RP4S2:  MOV     RP4DA1,R1       ;COMPUTE THE DISK
 9775  046466  005000                          CLR     R0
 9776  046470  071027  000630                  DIV     #408.,R0
 9777  046474  010137  046034                  MOV     R1,RP4DA1
 9778  046500  005037  046036                  CLR     RP4DA2
 9779  046504  000207                          RTS     PC
 9780
 9781  046506  005437  046044          RP4S3:  NEG     RP4WCT          ;COMPUTE VALID WORD COUNT
 9782  046512  042737  177700  046042          BIC     #177700,RP4MA2  ;AND MEMORY ADDRESS
 9783  046520  000207                          RTS     PC
 9784
 9785  046522  012737  000040  046024  RP4RDY: MOV     #BIT5,RP4USE    ;CLEAR CONTROLLER AND
 9786  046530  053737  046032  046024          BIS     RP4UNI,RP4USE
 9787  046536  013777  046024  135032          MOV     RP4USE,@RP4CS2
 9788  046544  013777  046032  135024          MOV     RP4UNI,@RP4CS2
 9789  046552  105777  135010          1$:     TSTB    @RP4CS1         ;DRIVES
 9790  046556  100375                          BPL     1$
 9791  046560  013777  046032  135010          MOV     RP4UNI,@RP4CS2
 9792  046566  012777  000021  134772          MOV     #21,@RP4CS1     ;INITIALIZE THE DRIVE
 9793  046574  017701  134766          2$:     MOV     @RP4CS1,R1      ;BY DOING A NOP
 9794  046600  005701                          TST     R1              ;WAIT FOR ERROR OR
 9795  046602  100431                          BMI     4$              ;READY
 9796  046604  105701                          TSTB    R1
 9797  046606  100372                          BPL     2$
 9798
 9799  046610  017700  134764          3$:     MOV     @RP4DS,R0       ;LOOK AT THE DRIVE
 9800                                                                  ;STATUS
 9801  046614  032700  000400                  BIT     #BIT8,R0        ;DRIVE PRESENT?
 9802  046620  001425                          BEQ     5$
 9803  046622  032700  000100                  BIT     #BIT6,R0        ;VOLUME VALID?
 9804  046626  001422                          BEQ     5$
 9805  046630  032700  010000                  BIT     #BIT12,R0       ;ON LINE?
 9806  046634  001417                          BEQ     5$
 9807  046636  032700  040000                  BIT     #BIT14,R0       ;ANY ERRORS?
 9808  046642  001014                          BNE     5$
 9809  046644  032700  004000                  BIT     #BIT11,R0       ;WRITE LOCKED
 9810  046650  001011                          BNE     5$
 9811  046652  105700                          TSTB    R0              ;WAIT FOR DRIVE READY
 9812  046654  100347                          BPL     2$
 9813
 9814  046656  012777  010000  134734          MOV     #BIT12,@RP40F   ;SET 16-BIT MODE
 9815  046664  000207                          RTS     PC              ;RETURN READY.
 9816  046666  032701  040000          4$:     BIT     #BIT14,R1       ;ATTENTION OR ERROR?
 9817  046672  001746                          BEQ     3$
 9818  046674  005726                  5$:     TST     (SP)+
 9819  046676  017737  134674  046016          MOV     @RP4CS2,RP4ER2  ;FLAG AND RECORD
 9820  046704  017737  134670  046020          MOV     @RP4DS,RP4ER3   ;ERROR
 9821  046712  017737  134664  046022          MOV     @RP4RR1,RP4ER4
 9822  046720  012737  177777  046014          MOV     #-1,RP4ER1
```

```
 9823  046726  004737  046736                  JSR     PC,RP4CLR       ;CLR THE CONTROLLER
 9824  046732  104406                          RESREG                  ;AND DRIVES.
 9825  046734  000002                          RTI                     ;RETURN
 9826
 9827  046736  013777  046024  134632  RP4CLR: MOV     PP4USE,@RP4CS2  ;CLR THE CONTROLLER
 9828  046744  105777  134616          1$:     TSTB    @RP4CS1         ;AND DRIVES.
 9829  046750  100375                          BPL     1$
 9830  046752  000207                          RTS     PC
 9831
 9832                                          .SBTTL          RS04 DISK HANDLE
 9833                                  ;RS04 DISK HANDLER
 9834
 9835                                  ;REGISTERS USED IN RS4HAN
 9836  046754  000000                  RS4FLG:.WORD    0
 9837  046756  000000                  RS4ER1:.WORD    0               ;ERROR FLAGS.
 9838  046760  000000                  RS4ER2:.WORD    0
 9839  046762  000000                  RS4ER3:.WORD    0
 9840  046764  000000                  RS4ER4:.WORD    0
 9841  046766  000000                  RS4USE:.WORD    0
 9842  046770  000000                  RS4TMP:.WORD    0
 9843  046772  000000                  RS4FUN:.WORD    0
 9844  046774  000000                  RS4UNI:.WORD    0
 9845  046776  000000                  RS4DA1:.WORD    0
 9846  047000  000000                  RS4DA2:.WORD    0
 9847  047002  000000                  RS4MA1:.WORD    0
 9848  047004  000000                  RS4MA2:.WORD    0
 9849  047006  000000                  RS4WCT:.WORD    0
 9850  047010  000000                  RS4VEC:.WORD    0
 9851  047012  000000                  RS4TRK:.WORD    0
 9852  047014  000000                  RS4SEC:.WORD    0
 9853  047016  000000                  RS4CYL:.WORD    0
 9854
 9855  047020  005737  046754          RS4HAN: TST     RS4FLG          ;SEE IF THERE ALREADY
 9856  047024  001402                          BEQ     RS4H1           ;IS AN RS04 FUNCTION
 9857  047026  104000                          ERROR                   ;IN PROGRESS. IF SO
 9858  047030  000000                          HALT                    ;ERROR. (SHOULD NEVER
 9859  047032  012737  000340  177776  RS4H1:  MOV     #340,@#PSW      ;HAPPEN.
 9860  047040  011637  046770                  MOV     (SP),RS4TMP
 9861  047044  062716  000016                  ADD     #16,(SP)
 9862  047050  104405                          SAVREG                  ;RAISE THE PRIORITY
 9863  047052  013700  046770                  MOV     RS4TMP,R0       ;GET A POINTER TO
 9864  047056  112037  046772                  MOVB    (R0)+,RS4FUN    ;FUNCTION
 9865  047062  112037  046774                  MOVB    (R0)+,RS4UNI    ;GET THE DRIVE NUMBER
 9866  047066  012037  046776                  MOV     (R0)+,RS4DA1    ;DISK ADDRESS
 9867  047072  012037  047000                  MOV     (R0)+,RS4DA2
 9868  047076  012037  047002                  MOV     (R0)+,RS4MA1    ;MEMORY ADDRESS
 9869  047102  012037  047004                  MOV     (R0)+,RS4MA2
 9870  047106  012037  047006                  MOV     (R0)+,RS4WCT    ;WORD COUNT
 9871  047112  012037  047010                  MOV     (R0)+,RS4VEC    ;INTERRUPT HANDLER ADDRESS
 9872  047116  005037  046756                  CLR     RS4ER1          ;CLEAR THE ERROR FLAGS
 9873  047122  005037  046760                  CLR     RS4ER2
 9874  047126  005037  046762                  CLR     RS4ER3
 9875
 9876  047132  004737  047406                  JSR     PC,RS4S1                ;SET UP UNIT (DRIVE) NUMBER
 9877  047136  004737  047500                  JSR     PC,RS4RDY       ;INITIALIZE DRIVE AND
 9878                                          ;CONTROLLER
```

```
 9879  047142  004737  047416                    JSR     PC,RS4S2        ;COMPUTE TRACK AND SECTOR
 9880  047146  004737  047464                    JSR     PC,RS4S3        ;COMPUTE WORD COUNT.
 9881
 9882  047152  013777  046774  134360  RS4H2:    MOV     RS4UNI,@RS4CS2  ;SET UP TH  ^ONTROL
 9883  047160  013777  047006  134344            MOV     RS4WCT,@RS4WC   ;AND DRIVE   GISTERS
 9884  047166  013777  047002  134340            MOV     RS4MA1,@RS4BA
 9885  047174  013777  047004  134356            MOV     RS4MA2,@RS4BAE
 9886  047202  013777  046776  134326            MOV     RS4DA1,@RS4DA
 9887  047210  013700  003740                    MOV     RS4V,R0
 9888  047214  012720  047266                    MOV     #RS4H4,(R0)+    ;SET THE INTERRUPT
 9889  047220  012710  000340                    MOV     #340,(R0)
 9890  047224  013700  046772                    MOV     RS4FUN,R0
 9891  047230  010037  046754                    MOV     R0,RS4FLG
 9892  047234  110077  134270                    MOVB    R0,@RS4CS1      ;LOAD THE FUNCTION AND GO.
 9893  047240  032700  000100                    BIT     #BIT6,R0        ;SEE IF AN INTERRUPT
 9894  047244  001402                            BEQ     RS4H3           ;IS TO BE EXPECTED.
 9895  047246  104406                            RESREG                  ;IF YES THEN RETURN
 9896  047250  000002                            RTI
 9897                                                                    ;IF NOT INTERRUPTING
 9898  047252  004737  047302          RS4H3:    JSR     PC,RS4H5        ;THEN WAIT FOR THE
 9899  047256  005037  046754                    CLR     RS4FLG          ;FUNCTION TO FINISH
 9900  047262  104406                            RESREG
 9901  047264  000002                            RTI
 9902
 9903  047266  005037  046754          RS4H4:    CLR     RS4FLG          ;WHEN THE INTERRUPT OCCURS.
 9904  047272  004737  047302                    JSR     PC,RS4H5        ;MAKE SURE THERE WERE
 9905  047276  000177  177506                    JMP     @RS4VEC         ;NO ERRORS BEFORE GOING
 9906                                                                    ;TO THE INTERRUPT
 9907                                                                    ;SERVICE ROUTINE.
 9908  047302  010046          RS4H5:    MOV     R0,-(SP)
 9909  047304  053777  046774  134226  RS4H51:  BIS     RS4UNI,@RS4CS2
 9910  047312  017700  134212                    MOV     @RS4CS1,R0
 9911  047316  005700                            TST     R0              ;SEE IF THE FUNCTION
 9912  047320  100023                            BPL     RS4H6           ;WAS COMPLETED WITHOUT
 9913  047322  032700  060000                    BIT     #60000,R0       ;ERRORS
 9914  047326  001420                            BEQ     RS4H6
 9915  047330  017737  134204  046760            MOV     @RS4CS2,RS4ER2  ;IF ERRORS OCCURRED
 9916  047336  017737  134200  046762            MOV     @RS4DS,RS4ER3   ;SET THE INDICATORS
 9917  047344  017737  134174  046764            MOV     @RS4ER,RS4ER4
 9918  047352  012737  177777  046756            MOV     #-1,RS4ER1
 9919  047360  004737  047672                    JSR     PC,RS4CLR       ;THEN CLEAR THE CONTROL
 9920  047364  012600                            MOV     (SP)+,R0
 9921  047366  000207                            RTS     PC              ;AND DRIVES
 9922  047370  105700          RS4H6:    TSTB    R0
 9923  047372  100344                            BPL     RS4H51          ;WAIT FOR READY OR
 9924  047374  105777  134142                    TSTB    @RS4DS          ;ERROR
 9925  047400  100341                            BPL     RS4H51
 9926  047402  012600                            MOV     (SP)+,R0
 9927  047404  000207                            RTS     PC
 9928
 9929  047406  042737  177770  046774  RS4S1:    BIC     #177770,RS4UNI  ;SET UP DRIVE NUMBER
 9930  047414  000207                            RTS     PC
 9931
 9932  047416  013701  046776          RS4S2:    MOV     RS4DA1,R1       ;COMPUTE A DISK
 9933  047422  005000                            CLR     R0              ;ADDRESS
 9934  047424  071027  007000                    DIV     #3584.,R0
```

```
9935  047430  005000                          CLR     R0
9936  047432  071027  000100                  DIV     #100,R0
9937  047436  010037  047012                  MOV     R0,RS4TRK
9938  047442  010137  047016                  MOV     R1,RS4CYL
9939  047446  000300                          SWAB    R0
9940  047450  006200                          ASR     R0
9941  047452  006200                          ASR     R0
9942  047454  050001                          BIS     R0,R1
9943  047456  010137  046776                  MOV     R1,RS4DA1
9944  047462  000207                          RTS     PC
9945
9946  047464  005437  047006          RS4S3:  NEG     RS4WCT          ;COMPUTE A VALID WORD
9947  047470  042737  177700  047004          BIC     #177700,RS4MA2  ;COUNT AND MEMORY
9948  047476  000207                          RTS     PC              ;ADDRESS
9949  047500  012737  000040  046766  RS4RDY: MOV     #BIT5,RS4USE    ;CLEAR CONTROLLER AND DRIVES
9950  047506  053737  046774  046766          BIS     RS4UNI,RS4USE
9951  047514  013777  046766  134016          MOV     RS4USE,@RS4CS2
9952  047522  013777  046774  134010          MOV     RS4UNI,@RS4CS2
9953  047530  105777  133774          1$:     TSTB    @RS4CS1
9954  047534  100375                          BPL     1$
9955  047536  013777  046774  133774          MOV     RS4UNI,@RS4CS2
9956  047544  012777  000001  133756          MOV     #1,@RS4CS1      ;INITIALIZE THE DRIVE
9957  047552  017701  123752          2$:     MOV     @RS4CS1,R1      ;BY DOING A NOP.
9958  047556  005701                          TST     R1
9959  047560  100420                          BMI     4$
9960  047562  105701                          TSTB    R1
9961  047564  100372                          BPL     2$
9962
9963  047566  017700  133750          3$:     MOV     @RS4DS,R0       ;LOOK AT THE DRIVE STATUS
9964  047572  032700  000400                  BIT     #BIT8,R0        ;DRIVE PRESENT?
9965  047576  001414                          BEQ     5$
9966  047600  032700  010000                  BIT     #BIT12,R0       ;ON LINE?
9967  047604  001411                          BEQ     5$
9968  047606  032700  004000                  BIT     #BIT11,R0       ;WRITE LOCKED?
9969  047612  001006                          BNE     5$
9970  047614  105700                          TSTB    R0              ;DRIVE READY?
9971  047616  100355                          BPL     2$
9972  047620  000207                          RTS     PC
9973  047622  032701  040000          4$:     BIT     #BIT14,R1       ;ATTENTION OR ERROR?
9974  047626  001757                          BEQ     3$
9975  047630  005726                  5$:     TST     (SP)+
9976  047632  017737  133702  046760          MOV     @RS4CS2,RS4ER2  ;FLAG AND RECORD THE
9977  047640  017737  133676  046762          MOV     @RS4DS,RS4ER3   ;ERROR
9978  047646  017737  133672  046764          MOV     @RS4ER,RS4ER4
9979  047654  012737  177777  046756          MOV     #-1,RS4ER1
9980  047662  004737  047672                  JSR     PC,RS4CLR       ;CLR THE CONTROLLER
9981  047666  104406                          RESREG                  ;AND DRIVES AND RETURN.
9982  047670  000002                          RTI
9983
9984  047672  013777  046766  133630  RS4CLR: MOV     RS4USE,@RS4CS1  ;CLR THE CONTROLLER
9985  047700  105777  133624          1$:     TSTB    @RS4CS1
9986  047704  100375                          BPL     1$
9987  047706  000207                          RTS     PC
9988
9989
9990                                  .SBTTL          RK05 DISK HANDLER
```

```
 9991                                   ;RK05 DISK HANDLER
 9992
 9993                                   ;REGISTERS USED IN RK5HAN
 9994  047710  000000                   RK5FLG:.WORD    0
 9995  047712  000000                   RK5ER1:.WORD    0               ;ERROR FLAGS.
 9996  047714  000000                   RK5ER2:.WORD    0
 9997  047716  000000                   RK5ER3:.WORD    0
 9998  047720  000000                   RK5ER4:.WORD    0
 9999  047722  000000                   RK5USE:.WORD    0
10000  047724  000000                   RK5TMP:.WORD    0
10001  047726  000000                   RK5FUN:.WORD    0
10002  047730  000000                   RK5UNI:.WORD    0
10003  047732  000000                   RK5DA1:.WORD    0
10004  047734  000000                   RK5DA2:.WORD    0
10005  047736  000000                   RK5MA1:.WORD    0
10006  047740  000000                   RK5MA2:.WORD    0
10007  047742  000000                   RK5WCT:.WORD    0
10008  047744  000000                   RK5VEC:.WORD    0
10009  047746  000000                   RK5TRK:.WORD    0
10010  047750  000000                   RK5SEC:.WORD    0
10011  047752  000000                   RK5CYL:.WORD    0
10012
10013  047754  005737  047710           RK5HAN: TST     RK5FLG          ;SEE IF THERE IS ALREADY AN
10014  047760  001402                          BEQ     RK5H1           ;RK05 FUNCTION IN PROGRESS
10015  047762  104000                          ERROR
10016  047764  000000                          HALT
10017
10018  047766  012737  000340  177776 RK5H1:  MOV     #340,@#PSW      ;RAISE THE PRIORITY
10019  047774  011637  047724                  MOV     (SP),RK5TMP
10020  050000  062716  000016                  ADD     #16.,(SP)
10021  050004  104405                          SAVREG
10022  050006  013700  047724                  MOV     RK5TMP,R0
10023  050012  112037  047726                  MOVB    (R0)+,RK5FUN    ;GET THE ARGUMENTS.
10024  050016  112037  047730                  MOVB    (R0)+,RK5UNI
10025  050022  012037  047732                  MOV     (R0)+,RK5DA1
10026  050026  012037  047734                  MOV     (R0)+,RK5DA2
10027  050032  012037  047736                  MOV     (R0)+,RK5MA1
10028  050036  012037  047740                  MOV     (R0)+,RK5MA2
10029  050042  012037  047742                  MOV     (R0)+,RK5WCT
10030  050046  012037  047744                  MOV     (R0)+,RK5VEC
10031
10032  050052  005037  047712                  CLR     RK5ER1          ;CLR THE ERROR FLAGS
10033  050056  005037  047714                  CLR     RK5ER2
10034  050062  005037  047716                  CLR     RK5ER3
10035
10036  050066  004737  050336                  JSR     PC,RK5S1        ;SET UP THE DRIVE NUMBER
10037  050072  004737  050542                  JSR     PC,RK5RDY       ;GET THE DEVICE AND CONTROL
10038                                          ;READY
10039  050076  004737  050360                  JSR     PC,RK5S2        ;COMPUTE THE SURFACE
10040                                          ;CYLINDER AND SECTOR
10041                                          ;ADDRESS.
10042  050102  004737  050462                  JSR     PC,RK5S3        ;SET UP A WORD COUNT,
10043                                          ;THE UNIBUS MAP
10044                                          ;AND BUS ADDRESS.
10045
10046  050106  005077  133576          RK5H2:  CLR     @RK5CS1
```

```
10047  050112  013777  047730  133576          MOV     RK5UNI,@RK5DA   ;SET THE DEVICE REGISTERS
10048  050120  013777  047742  133564          MOV     RK5WCT,@RK5WC   ;TO DO THE FUNCTION
10049  050126  013777  047736  133560          MOV     RK5MA1,@RK5BA
10050  050134  053777  047740  133546          BIS     RK5MA2,@RK5CS1
10051  050142  053777  047732  133546          BIS     RK5DA1,@RK5DA
10052  050150  013700  003746                  MOV     RK5V,R0         ;LOAD THE INTERRUPT VECTOR
10053  050154  012720  050226                  MOV     #RK5H4,(R0)+
10054  050160  012710  000340                  MOV     #340,(R0)
10055  050164  013700  047726                  MOV     RK5FUN,R0
10056  050170  010037  047710                  MOV     R0,RK5FLG
10057  050174  050077  133510                  BIS     R0,@RK5CS1      ;LOAD THE FUNCTION AND
10058                                                                  ;GO
10059
10060  050200  032700  000100                  BIT     #BIT6,R0        ;SEE IF THE FUNCTION WILL
10061  050204  001402                          BEQ     RK5H3           ;INTERRUPT WHEN DONE.
10062  050206  104406                          RESREG                  ;IF YES RETURN
10063  050210  000002                          RTI
10064
10065  050212  004737  050254          RK5H3:  JSR     PC,RK5H5        ;IF THE FUNCTION WAS
10066  050216  005037  047710                  CLR     RK5FLG          ;NOT INTERRUPT ENABLED
10067  050222  104406                          RESREG                  ;WAIT FOR DONE OR ERROR.
10068  050224  000002                          RTI
10069
10070  050226  004737  050254          RK5H4:  JSR     PC,RK5H5        ;SEE IF THERE WERE ANY ERRORS.
10071  050232  005037  047710                  CLR     RK5FLG
10072  050236  012777  050252  133502          MOV     #1$,@RK5V
10073  050244  000230                          SPL     0
10074  050246  000177  177472                  JMP     @RK5VEC
10075  050252  000002                  1$:     RTI
10076
10077  050254  010046                  RK5H5:  MOV     R0,-(SP)
10078  050256  017700  133426          RK5H51: MOV     @RK5CS1,R0      ;SEE IF ANY ERROR OCCURRED
10079  050262  005700                          TST     R0
10080  050264  100015                          BPL     RK5H6
10081  050266  017737  133414  047714          MOV     @RK5ER,RK5ER2   ;IF YES, FLAG THE ERROR
10082  050274  017737  133404  047716          MOV     @RK5DS,RK5ER3   ;AND SAVE THE STATUS
10083  050302  012737  177777  047712          MOV     #-1,RK5ER1
10084  050310  004737  050702                  JSR     PC,RK5CLR
10085  050314  012600                          MOV     (SP)+,R0
10086  050316  000207                          RTS     PC
10087
10088  050320  105700                  RK5H6:  TSTB    R0              ;WAIT FOR DONE OR
10089  050322  100355                          BPL     RK5H51          ;ERROR
10090  050324  105777  133354                  TSTB    @RK5DS
10091  050330  100352                          BPL     RK5H51
10092  050332  012600                          MOV     (SP)+,R0
10093  050334  000207                          RTS     PC
10094
10095  050336  013700  047730          RK5S1:  MOV     RK5UNI,R0
10096  050342  072027  000015                  ASH     #13.,R0
10097  050346  042700  017777                  BIC     #017777,R0
10098  050352  010037  047730                  MOV     R0,RK5UNI
10099  050356  000207                          RTS     PC
10100
10101  050360  013701  047732          RK5S2:  MOV     RK5DA1,R1       ;COMPUTE THE CYLINDER
10102  050364  005000                          CLR     R0              ;SURFACE AND SECTOR
```

```
10103  050366  071027  011100                DIV     #4672.,R0       ;DISK ADDRESS
10104  050372  005000                        CLR     R0
10105  050374  071027  000030                DIV     #24.,R0
10106  050400  010002                        MOV     R0,R2
10107  050402  005000                        CLR     R0
10108  050404  071027  000014                DIV     #12.,R0
10109  050410  010237  047752                MOV     R2,RK5CYL
10110  050414  010137  047750                MOV     R1,RK5SEC
10111  050420  010037  047746                MOV     R0,RK5TRK
10112  050424  072227  000005                ASH     #5,R2
10113  050430  042702  160037                BIC     #160037,R2
10114  050434  072027  000004                ASH     #4,R0
10115  050440  042700  177757                BIC     #177757,R0
10116  050444  042701  177760                BIC     #177760,R1
10117  050450  050100                        BIS     R1,R0
10118  050452  050200                        BIS     R2,R0
10119  050454  010037  047732                MOV     R0,RK5DA1
10120  050460  000207                        RTS     PC
10121
10122  050462  005437  047742        RK5S3:  NEG     RK5WCT          ;COMPUTE A VALID
10123                                                                ;WORD COUNT AND
10124  050466  013700  047736                MOV     RK5MA1,R0       ;SET THE UB MAP
10125  050472  013701  047740                MOV     RK5MA2,R1       ;REGISTERS
10126  050476  042701  177700                BIC     #177700,R1
10127  050502  012702  170300                MOV     #MAPL20,R2
10128  050506  012703  000010                MOV     #10,R3
10129  050512  010022                1$:     MOV     R0,(R2)+
10130  050514  010122                        MOV     R1,(R2)+
10131  050516  062700  020000                ADD     #20000,R0
10132  050522  005501                        ADC     R1
10133  050524  077306                        SOB     R3,1$
10134  050526  012737  000040  047740        MOV     #40,RK5MA2
10135  050534  005037  047736                CLR     RK5MA1
10136  050540  000207                        RTS     PC
10137
10138  050542  053777  047730  133146  RK5RDY: BIS   RK5UNI,@RK5DA   ;DO A CONTROL CLEAR
10139  050550  012777  000001  133132        MOV     #1,@RK5CS1      ;FUNCTION
10140  050556  105777  133126          1$:   TSTB    @RK5CS1
10141  050562  100375                        BPL     1$
10142
10143  050564  053777  047730  133124        BIS     RK5UNI,@RK5DA   ;DO A DRIVE CLEAR
10144  050572  012777  000015  133110        MOV     #15,@RK5CS1     ;FUNCTION
10145
10146  050600  017701  133104          2$:   MOV     @RK5CS1,R1      ;WAIT FOR DONE OR
10147  050604  100420                        BMI     5$              ;ERROR.
10148  050606  105701                        TSTB    R1
10149  050610  100373                        BPL     2$
10150
10151  050612  017701  133066          3$:   MOV     @RK5DS,R1
10152  050616  032701  000040                BIT     #BIT5,R1        ;WRITE ENABLED?
10153  050622  001011                        BNE     5$
10154  050624  005777  133056                TST     @RK5ER
10155  050630  100406                        BMI     5$
10156  050632  105701                        TSTB    R1
10157  050634  100366                        BPL     3$
10158  050636  032701  000100                BIT     #BIT6,R1
```

```
10159  050642  001763                        BEQ     3$
10160  050644  000207              4$:       RTS     PC
10161
10162  050646  005726              5$:       TST     (SP)+
10163  050650  017737  133032  047714        MOV     @RK5ER,RK5ER2
10164  050656  017737  133022  047716        MOV     @RK5DS,RK5ER3
10165  050664  012737  177777  047712        MOV     #-1,RK5ER1
10166  050672  004737  050702                JSR     PC,RK5CLR
10167  050676  104406                        RESREG
10168  050700  000002                        RTI
10169
10170  050702  005077  133010        RK5CLR: CLR     @RK5DA          ;RESET THE CONTROLLER
10171  050706  012777  000001  132774        MOV     #1,@RK5CS1      ;BY DOING A CONTROL
10172  050714  105777  132770        1$:       TSTB    @RK5CS1         ;CLEAR FUNCTION
10173  050720  100375                        BPL     1$
10174  050722  000207                        RTS     PC
10175
10176                                        .SBTTL          UNIBUS EXERCISER HANDLER
10177                                ;UNIBUS EXERCISER HANDLER
10178
10179                                ;REGISTERS USED IN UBEHAN
10180  050724  000000                UBEFLG:.WORD    0
10181  050726  000000                UBEER1:.WORD    0               ;ERROR FLAGS.
10182  050730  000000                UBEER2:.WORD    0
10183  050732  000000                UBEER3:.WORD    0
10184  050734  000000                UBEER4:.WORD    0
10185  050736  000000                UBEUSE:.WORD    0
10186  050740  000000                UBETMP:.WORD    0
10187  050742  000000                UBEFUN:.WORD    0
10188  050744  000000                UBEUNI:.WORD    0
10189  050746  000000                UBEDA1:.WORD    0
10190  050750  000000                UBEDA2:.WORD    0
10191  050752  000000                UBEMA1:.WORD    0
10192  050754  000000                UBEMA2:.WORD    0
10193  050756  000000                UBEWCT:.WORD    0
10194  050760  000000                UBEVEC:.WORD    0
10195  050762  000000                UBETRK:.WORD    0
10196  050764  000000                UBESEC:.WORD    0
10197  050766  000000                UBECYL:.WORD    0
10198
10199  050770  005737  050724        UBEHAN: TST     UBEFLG          ;SEE IF THERE IS ALREADY
10200  050774  001402                        BEQ     UBEH1           ;A UNIBUS EXERCISER FUNCTION
10201  050776  104000                        ERROR                   ;IN PROGRESS. IF THERE
10202  051000  000000                        HALT                    ;IS ERROR. (SHOULD NEVER HAPPEN)
10203
10204  051002  012737  000340  177776  UBEH1:  MOV     #340,@#PSW      ;RAISE THE PRIORITY
10205  051010  011637  050740                MOV     (SP),UBETMP     ;GET AN ARGUMENT POINTER
10206  051014  062716  000016                ADD     #16,(SP)
10207  051020  104405                        SAVREG
10208  051022  013700  050740                MOV     UBETMP,R0       ;RESET THE RETURN ADDRESS
10209
10210  051026  012037  050742                MOV     (R0)+,UBEFUN    ;GET THE ARGUMENTS.
10211  051032  012037  050746                MOV     (R0)+,UBEDA1
10212  051036  012037  050750                MOV     (R0)+,UBEDA2
10213  051042  012037  050752                MOV     (R0)+,UBEMA1
10214  051046  012037  050754                MOV     (R0)+,UBEMA2
```

```
10215  051052  012037  050756                  MOV     (R0)+,UBEWCT
10216  051056  012037  050760                  MOV     (R0)+,UBEVEC
10217  051062  005037  050726                  CLR     UBEER1          ;CLEAR THE ERROR FLAGS
10218  051066  005037  050730                  CLR     UBEER2
10219  051072  005037  050732                  CLR     UBEER3
10220  051076  004737  051366                  JSR     PC,UBERDY
10221  051102  004737  051304                  JSR     PC,UBES1                ;GO SET UP THE BUS
10222                                                                  ;ADDRESS AND UB MAP
10223
10224  051106  013777  050756  132612  UBEH2:  MOV     UBEWCT,@UBECC   ;SET THE DEVICE
10225  051114  013777  050752  132606          MOV     UBEMA1,@UBEBA   ;REGISTERS
10226  051122  053777  050754  132606          BIS     UBEMA2,@UBECR2
10227  051130  013777  050750  132566          MOV     UBEDA2,@UBEDB
10228  051136  013700  003750                  MOV     UBEV,R0
10229  051142  012720  051214                  MOV     #UBEH4,(R0)+
10230  051146  012710  000340                  MOV     #340,(R0)
10231  051152  013700  050742                  MOV     UBEFUN,R0
10232  051156  010037  050724                  MOV     R0,UBEFLG
10233  051162  010077  132544                  MOV     R0,@UBECR1      ;LOAD THE FUNCTION
10234  051166  032700  000100                  BIT     #BIT6,R0        ;SEE IF THE FUNCTION
10235  051172  001402                          BEQ     UBEH3           ;IS INTERRUPT ENABLED
10236  051174  104406                          RESREG                  ;IF YES RETURN
10237  051176  000002                          RTI
10238
10239  051200  004737  051230          UBEH3:  JSR     PC,UBEH5        ;IF NOT INTERRUPT ENABLED
10240  051204  005037  050724                  CLR     UBEFLG          ;WAIT FOR DONE OR
10241  051210  104406                          RESREG                  ;ERROR
10242  051212  000002                          RTI
10243
10244  051214  005037  050724          UBEH4:  CLR     UBEFLG          ;WHEN THE INTERRUPT
10245  051220  004737  051230                  JSR     PC,UBEH5        ;OCCURS SEE IF ANY ERRORS
10246  051224  000177  177530                  JMP     @UBEVEC         ;OCCURRED
10247
10248  051230  010046                  UBEH5:  MOV     R0,-(SP)
10249  051232  017700  132474          UBEH51: MOV     @UBECR1,R0      ;WAIT FOR DONE OR
10250  051236  005700                          TST     R0              ;ERROR
10251  051240  100015                          BPL     UBEH6
10252
10253  051242  017737  132464  050730          MOV     @UBECR1,UBEER2
10254  051250  017737  132462  050732          MOV     @UBECR2,UBEER3
10255  051256  012737  177777  050726          MOV     #-1,UBEER1
10256  051264  004737  051442                  JSR     PC,UBCLR
10257  051270  012600                          MOV     (SP)+,R0
10258  051272  000207                          RTS     PC
10259
10260  051274  105700                  UBEH6:  TSTB    R0
10261  051276  100355                          BPL     UBEH51
10262  051300  012600                          MOV     (SP)+,R0
10263  051302  000207                          RTS     PC
10264
10265  051304  013700  050752          UBES1:  MOV     UBEMA1,R0       ;SET UP THE BUS ADDRESS
10266  051310  013701  050754                  MOV     UBEMA2,R1       ;AND UB MAPPING BOX
10267  051314  042701  177700                  BIC     #177700,R1
10268  051320  012702  170200                  MOV     #MAPL00,R2
10269  051324  012703  000010                  MOV     #10,R3
10270
```

```
10271  051330  010022                  1$:     MOV     R0,(R2)+
10272  051332  010122                          MOV     R1,(R2)+
10273  051334  062700  020000                  ADD     #20000,R0
10274  051340  005501                          ADC     R1
10275  051342  077306                          SOB     R3,1$
10276
10277  051344  005037  050754                  CLR     UBEMA2
10278  051350  005037  050752                  CLR     UBEMA1
10279  051354  005137  050756                  COM     UBEWCT
10280  051360  005237  050756                  INC     UBEWCT
10281  051364  000207                          RTS     PC
10282
10283  051366  005077  132342          UBERDY: CLR     @UBECLR         ;TRY TO GET DEVICE
10284                                                                  ;READY
10285  051372  017700  132334          1$:     MOV     @UBECR1,R0
10286  051376  100403                          BMI     2$
10287  051400  105700                          TSTB    R0
10288  051402  100373                          BPL     1$
10289  051404  000207                          RTS     PC
10290
10291  051406  005726                  2$:     TST     (SP)+
10292  051410  017737  132316  050730          MOV     @UBECR1,UBEER2
10293  051416  017737  132314  050732          MOV     @UBECR2,UBEER3
10294  051424  012737  177777  050732          MOV     #-1,UBEER3
10295  051432  004737  051442                  JSR     PC,UBCLR
10296  051436  104406                          RESREG
10297  051440  000002  .                       RTI
10298
10299  051442  005077  132266          UBCLR:  CLR     @UBECLR         ;CLEAR THE DEVICE.
10300  051446  105777  132260          1$:     TSTB    @UBECR1
10301  051452  100375                          BPL     1$
10302  051454  000207                          RTS     PC
10303
10304                                          .SBTTL          MASS BUS TESTER HANDLER
10305                                  ;THIS CODE IS FOR HANDLING THE MASS BUS
10306                                  ;TESTED DEVICE.
10307
10308                                  ;REGISTERS USED IN RH4HAN
10309  051456  000000                  RH4FLG:.WORD    0
10310  051460  000000                  RH4ER1:.WORD    0               ;ERROR FLAGS.
10311  051462  000000                  RH4ER2:.WORD    0
10312  051464  000000                  RH4ER3:.WORD    0
10313  051466  000000                  RH4ER4:.WORD    0
10314  051470  000000                  RH4USE:.WORD    0
10315  051472  000000                  RH4TMP:.WORD    0
10316  051474  000000                  RH4FUN:.WORD    0
10317  051476  000000                  RH4UNI:.WORD    0
10318  051500  000000                  RH4DA1:.WORD    0
10319  051502  000000                  RH4DA2:.WORD    0
10320  051504  000000                  RH4MA1:.WORD    0
10321  051506  000000                  RH4MA2:.WORD    0
10322  051510  000000                  RH4WCT:.WORD    0
10323  051512  000000                  RH4VEC:.WORD    0
10324  051514  000000                  RH4TRK:.WORD    0
10325  051516  000000                  RH4SEC:.WORD    0
10326  051520  000000                  RH4CYL:.WORD    0
```

```
10327
10328  051522  005737  051456          RH4HAN: TST     RH4FLG          ;SEE IF A FUNCTION
10329  051526  001402                          BEQ     RH4H1           ;IS ALREADY ACTIVE IF
10330  051530  104000                          ERROR                   ;SO ERROR.
10331  051532  000000                          HALT
10332
10333  051534  012777  000340  126234  RH4H1:  MOV     #340,@PSW       ;RAISE THE PRIORITY
10334  051542  011637  051472                  MOV     (SP),RH4TMP
10335  051546  062716  000016                  ADD     #16,(SP)
10336  051552  104405                          SAVREG
10337  051554  013700  051472                  MOV     RH4TMP,R0       ;RESET THE RETURN
10338  051560  112037  051474                  MOVB    (R0)+,RH4FUN
10339  051564  112037  051476                  MOVB    (R0)+,RH4UNI
10340  051570  012037  051500                  MOV     (R0)+,RH4DA1
10341  051574  012037  051502                  MOV     (R0)+,RH4DA2
10342  051600  012037  051504                  MOV     (R0)+,RH4MA1
10343  051604  012037  051506                  MOV     (R0)+,RH4MA2
10344  051610  012037  051510                  MOV     (R0)+,RH4WCT
10345  051614  011037  051512                  MOV     (R0),RH4VEC
10346  051620  005037  051460                  CLR     RH4ER1          ;CLEAR THE ERROR FLAGS
10347  051624  005037  051462                  CLR     RH4ER2
10348  051630  005037  051464                  CLR     RH4ER3
10349  051634  004737  052114                  JSR     PC,RH4S1        ;SET UP THE UNIT NUMBER
10350  051640  004737  052140                  JSR     PC,RH4RDY       ;GET THE UNIT READY
10351  051644  004737  052124                  JSR     PC,RH4S2
10352
10353  051650  013777  051476  131776  RH4H2:  MOV     RH4UNI,@RH4CS2  ;SET THE CONTROL REGISTERS
10354  051656  013777  051510  131762          MOV     RH4WCT,@RH4WC   ;AND DEVICE REGISTERS
10355  051664  013777  051504  131756          MOV     RH4MA1,@RH4BA
10356  051672  013777  051506  131776          MOV     RH4MA2,@RH4AE
10357  051700  013777  051500  131756          MOV     RH4DA1,@RH4DR
10358  051706  012777  004000  131754          MOV     #4000,@RH4MR1
10359  051714  000240                          NOP
10360  051716  013700  003744                  MOV     RH4V,R0         ;VECTOR
10361  051722  012720  051774                  MOV     #RH4H4,(R0)+
10362  051726  012710  000340                  MOV     #340,(R0)
10363  051732  013700  051474                  MOV     RH4FUN,R0
10364  051736  010037  051456                  MOV     R0,RH4FLG       ;LOAD THE FUNCTION AND
10365  051742  110077  131676                  MOVB    R0,@RH4CS1      ;GO
10366  051746  032700  000100                  BIT     #BIT6,R0        ;SEE IF THIS FUNCTION
10367  051752  001402                          BEQ     RH4H3           ;WILL INTERRUPT WHEN DONE
10368  051754  104406                          RESREG                  ;IF YES RETURN TO CALL
10369  051756  000002                          RTI
10370
10371  051760  004737  052010          RH4H3:  JSR     PC,RH4H5        ;IF NOT INTERRUPT
10372  051764  005037  051456                  CLR     RH4FLG          ;ENABLED WAIT FOR
10373  051770  104406                          RESREG                  ;THE FUNCTION TO
10374  051772  000002                          RTI                     ;FINISH THEN RETURN.
10375
10376  051774  005037  051456          RH4H4:  CLR     RH4FLG          ;WHEN THE INTERRUPT
10377  052000  004737  052010                  JSR     PC,RH4H5        ;OCCURS CHECKS FOR
10378  052004  000177  177502                  JMP     @RH4VEC         ;ERRORS. THEN GO TO THE
10379                                                                  ;SPECIFIED SERVICE
10380                                                                  ;ROUTINE
10381
10382  052010  010046          RH4H5:  MOV     R0,-(SP)
```

```
10383  052012  053777  051476  131634  RH4HS1: BIS     RH4UNI,@RH4CS2
10384  052020  017700  131620          MOV     @RH4CS1,R0      ;SEE IF THE FUNCTION
10385  052024  005700                  TST     R0              ;WAS COMPLETED WITHOUT
10386  052026  100023                  BPL     RH4H6           ;ERRORS.
10387  052030  032700  060000          BIT     #60000,R0
10388  052034  001420                  BEQ     RH4H6
10389  052036  017737  131612  051462          MOV     @RH4CS2,RH4ER2  ;IF ERRORS OCCURRED
10390  052044  017737  131606  051464          MOV     @RH4ST,RH4ER3   ;SAVE STATUS AND SET
10391  052052  017737  131602  051466          MOV     @RH4ER,RH4ER4
10392  052060  012737  177777  051460          MOV     #-1,RH4ER1      ;ERROR FLAGS.
10393  052066  004737  052332                  JSR     PC,RH4CLR
10394  052072  012600                  MOV     (SP)+,R0
10395  052074  000207                  RTS     PC
10396
10397  052076  105700          RH4H6:  TSTB    R0              ;WAIT FOR READY OR
10398  052100  100344                  BPL     RH4HS1          ;ERROR
10399  052102  105777  131550          TSTB    @RH4ST
10400  052106  100341                  BPL     RH4HS1
10401  052110  012600                  MOV     (SP)+,R0
10402  052112  000207                  RTS     PC
10403
10404  052114  042737  177770  051476  RH4S1:  BIC     #177770,RH4UNI  ;SET UP THE DRIVE NUMBER
10405  052122  000207                  RTS     PC
10406
10407  052124  012737  000000  051502  RH4S2:  MOV     #0,RH4DA2       ;FOR DEBUG.
10408  052132  005437  051510          NEG     RH4WCT          ;SET UP WORD COUNT
10409  052136  000207                  RTS     PC
10410
10411  052140  012737  000040  051470  RH4RDY: MOV     #BIT5,RH4USE    ;CLR THE CONTROLLER
10412  052146  053737  051476  051470          BIS     RH4UNI,RH4USE
10413  052154  013777  051470  131472          MOV     RH4USE,@RH4CS2
10414  052162  013777  051476  131464          MOV     RH4UNI,@RH4CS2
10415  052170  105777  131450          1$:     TSTB    @RH4CS1 ;AND DRIVES
10416  052174  100375                  BPL     1$
10417  052176  013777  051476  131450          MOV     RH4UNI,@RH4CS2  ;DO A NOP FUNCTION
10418  052204  012777  000001  131432          MOV     #1,@RH4CS1      ;TO INITIALIZE THE
10419                                                          ;DRIVE
10420  052212  017701  131426          2$:     MOV     @RH4CS1,R1      ;WAIT FOR READY OR ERROR.
10421  052216  005701                  TST     R1
10422  052220  100420                  BMI     4$
10423  052222  105701                  TSTB    R1
10424  052224  100372                  BPL     2$
10425
10426  052226  017700  131424          3$:     MOV     @RH4ST,R0       ;LOOK AT THE UNIT STATUS
10427  052232  032700  000400          BIT     #BIT8,R0        ;UNIT PRESENT?
10428  052236  001414                  BEQ     5$
10429  052240  032700  010000          BIT     #BIT12,R0       ;ON LINE?
10430  052244  001411                  BEQ     5$
10431  052246  032700  040000          BIT     #BIT14,R0       ;ANY ERRORS?
10432  052252  001006                  BNE     5$
10433  052254  105700                  TSTB    R0              ;WAIT FOR UNIT READY
10434  052256  100355                  BPL     2$
10435  052260  000207                  RTS     PC
10436
10437  052262  032701  040000          4$:     BIT     #BIT14,R1       ;ATTENTION OR ERROR
10438  052266  001757                  BEQ     3$
```

```
10439  052270  005726                    5$:     TST     (SP)+           ;FLAG AND RECORD ERROR
10440  052272  017737  131356  051462            MOV     @RH4CS2,RH4ER2
10441  052300  017737  131352  051464            MOV     @RH4ST,RH4ER3
10442  052306  017737  131346  051466            MOV     @RH4ER,RH4ER4
10443  052314  012737  177777  051460            MOV     #-1,RH4ER1
10444  052322  004737  052332                    JSR     PC,RH4CLR
10445  052326  104406                            RESREG
10446  052330  000002                            RTI
10447
10448  052332  013777  051470  131314  RH4CLR: MOV     RH4USE,@RH4CS2  ;CLR THE CONTROLLER
10449  052340  105777  131300          1$:     TSTB    @RH4CS1         ;AND DRIVES.
10450  052344  100375                            BPL     1$
10451  052346  000207                            RTS     PC
10452                                ;SPECIAL MESSAGES:
10453
10454  052350  041536  000200          CONCMS: .ASCIZ  '^C'<CRLF>
10455
10456  052354  047515  044516  047524  MMESRS: .ASCIZ  'MONITOR (OR LOADER) RESTORED!'<CRLF>
10457  052362  020122  047450  020122
10458  052370  047514  042101  051105
10459  052376  020051  042522  052123
10460  052404  051117  042105  100041
10461  052412     000
10462
10463  052413     200  047520  042527  POWERM: .ASCIZ  <CRLF>'POWER FAILURE, PROGRAM RESTARTING'<CRLF><CRLF>
10464  052420  020122  040506  046111
10465  052426  051125  026105  050040
10466  052434  047522  051107  046501
10467  052442  051040  051505  040524
10468  052450  052122  047111  100107
10469  052456  000200
10470
10471  052460  000011                    $TAB:   .ASCIZ  <TAB>
10472
10473  052462  042600  050130  041505  MTA5:   .ASCII  <CRLF>'EXPECTED DATA:'<CRLF>
10474  052470  042524  020104  040504
10475  052476  040524  100072
10476  052502  051107  052517  020120          .ASCIZ  'GROUP 0.GROUP 1.MEM EV.'<TAB>'MEM ODD.'<CRLF>
10477  052510  027060  051107  052517
10478  052516  020120  027061  042515
10479  052524  020115  053105  004456
10480  052532  042515  020115  042117
10481  052540  027104  000200
10482
10483  052544  042200  052101  020101  MTA11:  .ASCII  <CRLF>'DATA WRITTEN.'<TAB>'TEST ADDR.'<TAB>'ERROR REG.'<CRLF>
10484  052552  051127  052111  042524
10485  052560  027116  052011  051505
10486  052566  020124  042101  051104
10487  052574  004456  051105  047522
10488  052602  020122  042522  027107
10489  052610     200
10490
10491  052611     040  047111  000040  MTA17:  .ASCIZ  ' IN '
10492
10493  052616  054105  042520  052103  MTB17:  .ASCIZ  'EXPECTED DATA:'<CRLF>
10494  052624  042105  042040  052101
```

```
10495  052632  035101  000200
10496
10497  052636  054502  042524  004456  MTC17:  .ASCIZ  'BYTE.'<TAB>
10498  052644     000
10499
10500  052645     127  051117  027104  MTA20:  .ASCIZ  'WORD.'<TAB>
10501  052652  000011
10502
10503  052654  054105  042520  052103  MTA21:  .ASCII  'EXPECTED DATA:'<CRLF>
10504  052662  042105  042040  052101
10505  052670  035101     200
10506  052673     110  052111  020123          .ASCIZ  'HITS IN GROUP 0.'<TAB>'/'<TAB>'HITS IN GROUP 1. '<CRLF>
10507  052700  047111  043440  047522
10508  052706  050125  030040  004456
10509  052714  004457  044510  051524
10510  052722  044440  020116  051107
10511  052730  052517  020120  027061
10512  052736  100040     000
10513
10514          052611                  MTB21=MTA17
10515
10516  052741     200  042524  052123  MTA43:  .ASCII  <CRLF>'TEST ADDRESS.'<TAB>'ERROR ADRS REG.'<TAB>
10517  052746  040440  042104  042522
10518  052754  051523  004456  051105
10519  052762  047522  020122  042101
10520  052770  051522  051040  043505
10521  052776  004456
10522  053000  051105  047522  020122          .ASCIZ  'ERROR REG.'<CRLF>
10523  053006  042522  027107  000200
10524
10525  053014  053600  047522  042524  MTA45:  .ASCIZ  <CRLF>'WROTE. 377'<TAB>'IN BYTE. '
10526  053022  020056  033463  004467
10527  053030  047111  041040  052131
10528  053036  027105  000040
10529
10530  053042  051200  040505  020104  MTB45:  .ASCIZ  <CRLF>'READ DATA. '
10531  053050  040504  040524  020056
10532  053056     000
10533
10534  053057     011  047111  053440  MTC45:  .ASCIZ  <TAB>'IN WORD. '
10535  053064  051117  027104  000040
10536
10537  053072  053600  047522  042524  MTA50:  .ASCIZ  <CRLF>'WROTE. 000'<TAB>'IN BYTE. '
10538  053100  020056  030060  004460
10539  053106  047111  041040  052131
10540  053114  027105  000040
10541
10542  053120  042600  052116  051105  PDMSG1: .ASCII  <CRLF>'ENTERING CACHE ADDRESS MEMORY POWER UP '
10543  053126  047111  020107  040503
10544  053134  044103  020105  042101
10545  053142  051104  051505  020123
10546  053150  042515  047515  054522
10547  053156  050040  053517  051105
10548  053164  052440  020120
10549  053170  047111  040526  044514          .ASCII  'INVALIDATOR TEST.'<CRLF>
10550  053176  040504  047524  020122
```

```
10551  053204  042524  052123  100056
10552  053212  046120  040505  042523          .ASCII  'PLEASE GO THROUGH A POWER DOWN, POWER UP '
10553  053220  043440  020117  044124
10554  053226  047522  043525  020110
10555  053234  020101  047520  042527
10556  053242  020122  047504  047127
10557  053250  020054  047520  042527
10558  053256  020122  050125     040
10559  053263     123  050505  042525          .ASCIZ  'SEQUENCE.'<CRLF>
10560  053270  041516  027105  000200
10561
10562  053276  041600  041501  042510  PDMSG2: .ASCII  <CRLF>'CACHE ADDRESS MEMORY POWER UP INVALIDATOR'
10563  053304  040440  042104  042522
10564  053312  051523  046440  046505
10565  053320  051117  020131  047520
10566  053326  042527  020122  050125
10567  053334  044440  053116  046101
10568  053342  042111  052101  051117
10569  053350  052040  051505  020124          .ASCIZ  ' TEST DID NOT FAIL.'<CRLF>
10570  053356  044504  020104  047516
10571  053364  020124  040506  046111
10572  053372  100056     000
10573
10574  053375     105  051122  051117  ADRNG:  .ASCII  'ERROR ADDRESS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
10575  053402  040440  042104  042522
10576  053410  051523  051040  043505
10577  053416  051511  042524  020122
10578  053424  042516  042105  042105
10579  053432  043040  051117  052040
10580  053440  051505  026124  041200
10581  053446  052125  044440  020124
10582  053454  040510  020123  042502
10583  053462  047105     040
10584  053465     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
10585  053472  042105  040440  020123
10586  053500  040502  020504     000
10587
10588  053505     105  051122  051117  ERRNG:  .ASCII  'ERROR REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
10589  053512  051040  043505  051511
10590  053520  042524  020122  042516
10591  053526  042105  042105  043040
10592  053534  051117  052040  051505
10593  053542  026124  041200  052125
10594  053550  044440  020124  040510
10595  053556  020123  042502  047105
10596  053564     040
10597  053565     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
10598  053572  042105  040440  020123
10599  053600  040502  020504     000
10600
10601  053605     103  047117  051124  CNRNG:  .ASCII  'CONTROL REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
10602  053612  046117  051040  043505
10603  053620  051511  042524  020122
10604  053626  042516  042105  042105
10605  053634  043040  051117  052040
10606  053642  051505  026124  041200
```

```
10627  053650  052125  044440  020124
10628  053656  040510  020123  042502
10629  053664  047105     040
10630  053667     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
10631  053674  042105  040440  020123
10632  053702  040502  020504     000
10633  053707     115  044501  052116  MNRNG:  .ASCII  'MAINTENANCE REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
10634  053714  047105  047101  042503
10635  053722  051040  043505  051511
10636  053730  042524  020122  042516
10637  053736  042105  042105  043040
10638  053744  051117  052040  051505
10639  053752  026124  041200  052125
10640  053760  044440  020124  040510
10641  053766  020123  042502  047105
10642  053774     040
10643  053775     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
10644  054002  042105  040440  020123
10645  054010  040502  020504     000
10646
10647  054015     110  052111  046457  HMRNG:  .ASCII  'HIT/MISS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
10648  054022  051511  020123  042522
10649  054030  044507  052123  051105
10650  054036  047040  042505  042504
10651  054044  020104  047506  020122
10652  054052  042524  052123  100054
10653  054060  052502  020124  052111
10654  054066  044040  051501  041040
10655  054074  042505  020116
10656  054100  046106  043501  042507          .ASCIZ  'FLAGGED AS BAD!'
10657  054106  020104  051501  041040
10658  054114  042101  000041
10659
10660  054120  040600  042104  042522  MTA77:  .ASCIZ  <CRLF>'ADDRESS:  '
10661  054126  051523  020072  000040
10662
10663  054134  051440  047510  046125  MTB77:  .ASCIZ  ' SHOULD HAVE BEEN A HIT IN GROUP '
10664  054142  020104  040510  042526
10665  054150  041040  042505  020116
10666  054156  020101  044510  020124
10667  054164  047111  043440  047522
10668  054172  050125  000040
10669
10670  054176  043101  042524  020122  MTC77:  .ASCIZ  'AFTER REFERENCING'<CRLF>'ADDRESS:  '
10671  054204  042522  042506  042522
10672  054212  041516  047111  100107
10673  054220  042101  051104  051505
10674  054226  035123  020040     000
10675
10676  054233     040  044127  046111  MTD77:  .ASCIZ  ' WHILE FORCING SELECTION OF GROUP '
10677  054240  020105  047506  041522
10678  054246  047111  020107  042523
10679  054254  042514  052103  047511
10680  054262  020116  043117  043440
10681  054270  047522  050125  000040
10682
```

```
10663  054276  040600  051122  051117  MTA101: .ASCII  <CRLF>'ARROR ADRS REG.'<TAB>'ERROR REG.'<TAB>
10664  054304  040440  051104  020123
10665  054312  042522  027107  042411
10666  054320  051122  051117  051040
10667  054326  043505  004456
10668  054332  054105  042520  052103          .ASCIZ  'EXPECTED ERR.'<TAB>'PATTERN PUT IN MAINT REG.'<CRLF>
10669  054340  042105  042440  051122
10670  054346  004456  040520  052124
10671  054354  051105  020116  052520
10672  054362  020124  047111  046440
10673  054370  044501  052116  051040
10674  054376  043505  100056     000
10675
10676  054403     200  043101  042524  MTA120: .ASCIZ  <CRLF>'AFTER 2ND CYCLE READ  '
10677  054410  020122  047062  020104
10678  054416  054503  046103  020105
10679  054424  042522  042101  020040
10680  054432     000
10681
10682  054433     200  043101  042524  MTB120: .ASCIZ  <CRLF>'AFTER 4TH CYCLE READ  '
10683  054440  020122  052064  020110
10684  054446  054503  046103  020105
10685  054454  042522  042101  020040
10686  054462     000
10687
10688  054463     200  043101  042524  MTC120: .ASCIZ  <CRLF>'AFTER 6TH CYCLE READ  '
10689  054470  020122  052066  020110
10690  054476  054503  046103  020105
10691  054504  042522  042101  020040
10692  054512     000
10693  054513     200  043101  042524  MTD120: .ASCIZ  <CRLF>'AFTER 8TH CYCLE READ  '
10694  054520  020122  052070  020110
10695  054526  054503  046103  020105
10696  054534  042522  042101  020040
10697  054542     000
10698
10699  054543     200  043101  042524  MTE120: .ASCIZ  <CRLF>'AFTER 10TH CYCLE READ '
10700  054550  020122  030061  044124
10701  054556  041440  041531  042514
10702  054564  051040  040505  020104
10703  054572     000
10704
10705  054573     200  043101  042524  MTF120: .ASCIZ  <CRLF>'AFTER 12TH CYCLE READ '
10706  054600  020122  031061  044124
10707  054606  041440  041531  042514
10708  054614  051040  040505  020104
10709  054622     000
10710
10711  054623     106  047522  020115  MTG120: .ASCIZ  'FROM THE HIT/MISS REG. EXPECTED '
10712  054630  044124  020105  044510
10713  054636  027524  044515  051523
10714  054644  051040  043505  020056
10715  054652  054105  042520  052103
10716  054660  042105  000040
10717
10718  054664  052200  042510  050040  MTA124: .ASCII  <CRLF>'THE PATTERN BEING USED IN THE MAINTENANCE '
```

```
 10719  054672  052101  042524  047122
 10720  054700  041040  044505  043516
 10721  054706  052440  042523  020104
 10722  054714  047111  052040  042510
 10723  054722  046440  044501  052116
 10724  054730  047105  047101  042503
 10725  054736     040
 10726  054737     122  043505  051511          .ASCIZ  'REGISTER WAS: '
 10727  054744  042524  020122  040527
 10728  054752  035123  000040
 10729
 10730  054756  051200  043105  051105  MTA126: .ASCIZ  <CRLF>'REFERENCED ADDRESS:'<TAB>
 10731  054764  047105  042503  020104
 10732  054772  042101  051104  051505
 10733  055000  035123  000011
 10734
 10735  055004  040600  051122  051117  MTB126: .ASCIZ  <CRLF>'ARROR ADDRESS REGISTER:'<TAB>
 10736  055012  040440  042104  042522
 10737  055020  051523  051040  043505
 10738  055026  051511  042524  035122
 10739  055034  000011
 10740
 10741  055036  050200  052101  042524  MTA131: .ASCIZ  <CRLF>'PATTERN BEING USED IN THE MAINTENANCE REGISTER:'<TAB>
 10742  055044  047122  041040  044505
 10743  055052  043516  052440  042523
 10744  055060  020104  047111  052040
 10745  055066  042510  046440  044501
 10746  055074  052116  047105  047101
 10747  055102  042503  051040  043505
 10748  055110  051511  042524  035122
 10749  055116  000011
 10750
 10751  055120  042600  050130  041505  MTB131: .ASCIZ  <CRLF>'EXPECTED ERROR REGISTER:'<TAB>
 10752  055126  042524  020104  051105
 10753  055134  047522  020122  042522
 10754  055142  044507  052123  051105
 10755  055150  004472     000
 10756
 10757  055153     200  047507  020124  MTC131: .ASCIZ  <CRLF>'GOT ERROR REGISTER:'<TAB>
 10758  055160  051105  047522  020122
 10759  055166  042522  044507  052123
 10760  055174  051105  004472     000
 10761
 10762  055201     200  051105  047522  MTA134: .ASCIZ  <CRLF>'ERROR HDR REG.'<TAB>'ERROR REG.'<CRLF>
 10763  055206  020122  042101  020122
 10764  055214  042522  027107  042411
 10765  055222  051122  051117  051040
 10766  055230  043505  100056     000
 10767
 10768  055235     200  054105  042520  MTA135: .ASCIZ  <CRLF>'EXPECTED ERROR REG.:  '
 10769  055242  052103  042105  042440
 10770  055250  051122  051117  051040
 10771  055256  043505  035056  020040
 10772  055264     000
 10773
 10774  055265     107  052117  042440  MTB135: .ASCIZ  'GOT ERROR REG.:  '
```

```
10775  055272  051122  051117  051040
10776  055300  043505  035056  020040
10777  055306     000
10778
10779  055307     200  054105  042520  MTC135: .ASCIZ  <CRLF>'EXPECTED ERROR ADR REG.:  '
10780  055314  052103  042105  042440
10781  055322  051122  051117  040440
10782  055330  051104  051040  043505
10783  055336  035056  020040     000
10784
10785  055343     107  052117  042440  MTD135: .ASCIZ  'GOT ERROR ADR REG.:  '
10786  055350  051122  051117  040440
10787  055356  051104  051040  043505
10788  055364  035056  020040     000
10789
10790
10791                                  ;THESE ARE THE ERROR MESSAGES:
10792
10793  055371     101  051040  043105  EM1:    .ASCIZ  'A REFERENCE WHICH SHOULD HAVE BEEN A HIT WAS A MISS.'
10794  055376  051105  047105  042503
10795  055404  053440  044510  044103
10796  055412  051440  047510  046125
10797  055420  020104  040510  042526
10798  055426  041040  042505  020116
10799  055434  020101  044510  020124
10800  055442  040527  020123  020101
10801  055450  044515  051523  000056
10802
10803  055456  047125  054105  042520  EM2:    .ASCII  'UNEXPECTED ERROR DURING WORST CASE NOISE TEST ON '
10804  055464  052103  042105  042440
10805  055472  051122  051117  042040
10806  055500  051125  047111  020107
10807  055506  047527  051522  020124
10808  055514  040503  042523  047040
10809  055522  044517  042523  052040
10810  055530  051505  020124  047117
10811  055536     040
10812  055537     103  041501  042510          .ASCII  'CACHE DATA MEMORY.'<CRLF>
10813  055544  042040  052101  020101
10814  055552  042515  047515  054522
10815  055560  100056
10816  055562  020101  047516  026516          .ASCIZ  'A NON-CACHE DATA PARITY ERROR OCCURRED WHILE TESTING.'
10817  055570  040503  044103  020105
10818  055576  040504  040524  050040
10819  055604  051101  052111  020131
10820  055612  051105  047522  020122
10821  055620  041517  052503  051122
10822  055626  042105  053440  044510
10823  055634  042514  052040  051505
10824  055642  044524  043516  000056
10825
10826  055650  047527  051522  020124  EM3:    .ASCII  'WORST CASE NOISE TEST OF THE CACHE DATA MEMORY '
10827  055656  040503  042523  047040
10828  055664  044517  042523  052040
10829  055672  051505  020124  043117
10830  055700  052040  042510  041440
```

```
10831  055706  041501  042510  042040
10832  055714  052101  020101  042515
10833  055722  047515  054522     040
10834  055727     200  040506  046111          .ASCIZ  <CRLF>/FAILED WHILE GALLOPING 0'S./
10835  055734  042105  053440  044510
10836  055742  042514  043440  046101
10837  055750  047514  044520  043516
10838  055756  030040  051447  000056
10839
10840  055764  047527  051522  020124  EM4:    .ASCII  'WORST CASE NOISE TEST OF THE CACHE DATA MEMORY'
10841  055772  040503  042523  047040
10842  056000  044517  042523  052040
10843  056006  051505  020124  043117
10844  056014  052040  042510  041440
10845  056022  041501  042510  042040
10846  056030  052101  020101  042515
10847  056036  047515  054522
10848  056042  043200  044501  042514          .ASCIZ  <CRLF>/FAILED WHILE GALLOPING 1'S./
10849  056050  020104  044127  046111
10850  056056  020105  040507  046114
10851  056064  050117  047111  020107
10852  056072  023461  027123     000
10853
10854  056077     103  046504  020130  EM5:    .ASCIZ  'CDMX TEST FAILURE.'<CRLF>'BAD CACHE GROUP 0 DATA READ.'
10855  056104  042524  052123  043040
10856  056112  044501  052514  042522
10857  056120  100056  040502  020104
10858  056126  040503  044103  020105
10859  056134  051107  052517  020120
10860  056142  020060  040504  040524
10861  056150  051040  040505  027104
10862  056156     000
10863
10864  056157     103  046504  020130  EM6:    .ASCIZ  'CDMX TEST FAILURE.'<CRLF>'BAD CACHE GROUP 1 DATA READ.'
10865  056164  042524  052123  043040
10866  056172  044501  052514  042522
10867  056200  100056  040502  020104
10868  056206  040503  044103  020105
10869  056214  051107  052517  020120
10870  056222  020061  040504  040524
10871  056230  051040  040505  027104
10872  056236     000
10873
10874  056237     103  046504  020130  EM7:    .ASCII  'CDMX TEST FAILURE.'<CRLF>'BAD MAIN MEMORY, EVEN WORD,'
10875  056244  042524  052123  043040
10876  056252  044501  052514  042522
10877  056260  100056  040502  020104
10878  056266  040515  047111  046440
10879  056274  046505  051117  026131
10880  056302  042440  042526  020116
10881  056310  047527  042122     054
10882  056315     040  040504  040524          .ASCIZ  ' DATA READ.'
10883  056322  051040  040505  027104
10884  056330     000
10885
10886  056331     103  046504  020130  EM10:   .ASCII  'CDMX TEST FAILURE.'<CRLF>'BAD MAIN MEMORY, ODD WORD,'
```

```
 10887  056336  042524  052123  043040
 10888  056344  044501  052514  042522
 10889  056352  100056  040502  020104
 10890  056360  040515  047111  046440
 10891  056366  046505  051117  026131
 10892  056374  047440  042104  053440
 10893  056402  051117  026104
 10894  056406  042040  052101  020101          .ASCIZ  ' DATA READ.'
 10895  056414  042522  042101  000056
 10896
 10897  056422  040520  044522  054524  EM11:   .ASCIZ  'PARITY ERROR IN CACHE DATA MEMORY COUNT PATTERN TEST.'
 10898  056430  042440  051122  051117
 10899  056436  044440  020116  040503
 10900  056444  044103  020105  040504
 10901  056452  040524  046440  046505
 10902  056460  051117  020131  047503
 10903  056466  047125  020124  040520
 10904  056474  052124  020105  020116
 10905  056502  042524  000123  000056
 10906
 10907  056510  040502  020104  040504  EM12:   .ASCII  'BAD DATA WAS READ IN CACHE MEMORY COUNT PATTERN '
 10908  056516  040524  053440  051501
 10909  056524  051040  040505  020104
 10910  056532  047111  041440  041501
 10911  056540  042510  046440  046505
 10912  056546  051117  020131  047503
 10913  056554  047125  020124  040520
 10914  056562  052124  051105  020116
 10915  056570  042524  052123  100056          .ASCIZ  'TEST.'<CRLF>'BUT NO TRAP OR ABORT OCCURRED.'
 10916  056576  052502  020124  047516
 10917  056604  052040  040522  020120
 10918  056612  051117  040440  047502
 10919  056620  052122  047440  041503
 10920  056626  051125  042522  027104
 10921  056634     000
 10922
 10923  056635     103  041501  042510  EM13:   .ASCII  'CACHE MEMORY COUNT PATTERN TEST.'<CRLF>
 10924  056642  046440  046505  051117
 10925  056650  020131  047503  047125
 10926  056656  020124  040520  052124
 10927  056664  051105  020116  042524
 10928  056672  052123  100056
 10929  056676  051105  047522  020122          .ASCIZ  'ERROR SUMMARY.'
 10930  056704  052523  046515  051101
 10931  056712  027131     000
 10932
 10933  056715     200  047125  054105  EM14:   .ASCIZ  <CRLF>'UNEXPECTED PARITY ERROR TRAP.'
 10934  056722  042520  052103  042105
 10935  056730  050040  051101  052111
 10936  056736  020131  051105  047522
 10937  056744  020122  051124  050101
 10938  056752  000056
 10939
 10940  056754  025052  052052  051505  EM15:   .ASCIZ  '***TEST ABORTED! GOING TO NEXT TEST.***'
 10941  056762  020124  041101  051117
 10942  056770  042524  020504  043440
```

```
10943  056776  044517  043516  052040
10944  057004  020117  042516  052130
10945  057012  052040  051505  027124
10946  057020  025052  000052
10947
10948  057024  040503  044103  020105  EM16:   .ASCIZ  'CACHE DATA MEMORY DUAL ADDRESS TEST FAILED.'
10949  057032  040504  040524  046440
10950  057040  046505  051117  020131
10951  057046  052504  046101  040440
10952  057054  042104  042522  051523
10953  057062  052040  051505  020124
10954  057070  040506  046111  042105
10955  057076  000056
10956
10957  057100  040503  044103  020105  EM17:   .ASCIZ  'CACHE DATA MEMORY BYTE ENABLE LOGIC TEST FAILED.'
10958  057106  040504  040524  046440
10959  057114  046505  051117  020131
10960  057122  054502  042524  042440
10961  057130  040516  046102  020105
10962  057136  047514  044507  020103
10963  057144  042524  052123  043040
10964  057152  044501  042514  027104
10965  057160     000
10966
10967          057100                  EM20=EM17
10968
10969  057161     103  041501  042510  EM21:   .ASCIZ  'CACHE DATA MEMORY CHIP SELECTION LOGIC TEST FAILED.'
10970  057166  042040  052101  020101
10971  057174  042515  047515  054522
10972  057202  041440  044510  020120
10973  057210  042523  042514  052103
10974  057216  047511  020116  047514
10975  057224  044507  020103  042524
10976  057232  052123  043040  044501
10977  057240  042514  027104     000
10978
10979  057245     101  042104  042522  EM22:   .ASCII  'ADDRESS MULTIPLEXER TEST WAS UNABLE TO FORCE'
10980  057252  051523  046440  046125
10981  057260  044524  046120  054105
10982  057266  051105  052040  051505
10983  057274  020124  040527  020123
10984  057302  047125  041101  042514
10985  057310  052040  020117  047506
10986  057316  041522     105
10987  057321     040  020101  040520          .ASCII  ' A PARITY ERROR, USING THE '<CRLF>
10988  057326  044522  054524  042440
10989  057334  051122  051117  020054
10990  057342  051525  047111  020107
10991  057350  044124  020105     200
10992  057355     115  044501  052116          .ASCII  'MAINTENANCE REGISTER, ON THE'
10993  057362  047105  047101  042503
10994  057370  051040  043505  051511
10995  057376  042524  026122  047440
10996  057404  020116  044124     105
10997  057411     040  040515  047111          .ASCIZ  ' MAIN MEMORY ADDRESS AND CONTROL LINES.'
10998  057416  046440  046505  051117
```

```
10999  057424  020131  042101  051104
11000  057432  051505  020123  047101
11001  057440  020104  047503  052116
11002  057446  047522  020114  044514
11003  057454  042516  027123     000
11004
11005  057461     101  042104  042522  EM23:   .ASCII  'ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FAILED.'
11006  057466  051523  046440  046125
11007  057474  044524  046120  054105
11008  057502  051105  020054  046501
11009  057510  026130  041440  052520
11010  057516  044440  050116  052125
11011  057524  020123  042524  052123
11012  057532  043040  044501  042514
11013  057540  027104
11014  057542  042600  051122  051117          .ASCIZ  <CRLF>'ERROR ADDRESS REGISTER NOT SET CORRECTLY.'
11015  057550  040440  042104  042522
11016  057556  051523  051040  043505
11017  057564  051511  042524  020122
11018  057572  047516  020124  042523
11019  057600  020124  047503  051122
11020  057606  041505  046124  027131
11021  057614     000
11022
11023          057245                  EM24=EM22
11024
11025          057461                  EM25=EM23
11026
11027  057615     101  042104  042522  EM26:   .ASCII  'ADDRESS MEMORY, ADDRESS COMPARATOR TEST FAILURE.'
11028  057622  051523  046440  046505
11029  057630  051117  026131  040440
11030  057636  042104  042522  051523
11031  057644  041440  046517  040520
11032  057652  040522  047524  020122
11033  057660  042524  052123  043040
11034  057666  044501  052514  042522
11035  057674     056
11036  057675     200  047101  040440          .ASCII  <CRLF>'AN ADDRESS WHICH SHOULD HAVE BEEN A HIT WAS'
11037  057702  042104  042522  051523
11038  057710  053440  044510  044103
11039  057716  051440  047510  046125
11040  057724  020104  040510  042526
11041  057732  041040  042505  020116
11042  057740  020101  044510  020124
11043  057746  040527     123
11044  057751     040  020101  044515          .ASCIZ  ' A MISS.'
11045  057756  051523  000056
11046
11047  057762  042101  051104  051505  EM27:   .ASCII  'ADDRESS MEMORY, ADDRESS COMPARATOR TEST FAILURE.'
11048  057770  020123  042515  047515
11049  057776  054522  020054  042101
11050  060004  051104  051505  020123
11051  060012  047503  050115  051101
11052  060020  052101  051117  052040
11053  060026  051505  020124  040506
11054  060034  046111  051125  027105
```

```
11055  060042  040600  020116  042101          .ASCII  <CRLF>'AN ADDRESS WHICH SHOULD HAVE BEEN A MISS '
11056  060050  051104  051505  020123
11057  060056  044127  041511  020110
11058  060064  044123  052517  042114
11059  060072  044040  053101  020105
11060  060100  042502  047105  040440
11061  060106  046440  051511  020123
11062  060114  040527  020123  020101          .ASCIZ  'WAS A HIT.'
11063  060122  044510  027124     000
11064
11065          057245                  EM30=EM22
11066
11067  060127     101  042104  042522  EM31:   .ASCII  'ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FAILED.'
11068  060134  051523  046440  046125
11069  060142  044524  046120  054105
11070  060150  051105  020054  046501
11071  060156  026130  052440  044516
11072  060164  052502  020123  047111
11073  060172  052520  051524  052040
11074  060200  051505  020124  040506
11075  060206  046111  042105     056
11076  060213     200  051105  047522          .ASCIZ  <CRLF>'ERROR ADDRESS REGISTER NOT SET CORRECTLY.'
11077  060220  020122  042101  051104
11078  060226  051505  020123  042522
11079  060234  044507  052123  051105
11080  060242  047040  052117  051440
11081  060250  052105  041440  051117
11082  060256  042522  052103  054514
11083  060264  000056
11084
11085          057245                  EM32=EM22
11086
11087          060127                  EM33=EM31
11088
11089  060266  042101  051104  051505  EM34:   .ASCII  'ADDRESS MULTIPLEXER, AMX, DUAL ADDRESS TEST,'<CRLF>
11090  060274  020123  052515  052114
11091  060302  050111  042514  042530
11092  060310  026122  040440  054115
11093  060316  020054  052504  046101
11094  060324  040440  042104  042522
11095  060332  051523  052040  051505
11096  060340  026124     200
11097  060343     117  020116  050103          .ASCIZ  'ON CPU INPUTS, FAILED.'
11098  060350  020125  047111  052520
11099  060356  051524  020054  040506
11100  060364  046111  042105  000056
11101
11102  060372  042101  051104  051505  EM35:   .ASCII  'ADDRESS MULTIPLEXER, AMX, DUAL ADDRESS TEST,'<CRLF>
11103  060400  020123  052515  052114
11104  060406  050111  042514  042530
11105  060414  026122  040440  054115
11106  060422  020054  052504  046101
11107  060430  040440  042104  042522
11108  060436  051523  052040  051505
11109  060444  026124     200
11110  060447     117  020116  047125          .ASCIZ  'ON UNIBUS INPUTS, FAILED.'
```

```
11111  060454  041111  051525  044440
11112  060462  050116  052125  026123
11113  060470  043040  044501  042514
11114  060476  027104     000
11115
11116  060501     101  042104  042522  EM36:   .ASCII  'ADDRESS MEMORY COUNT PATTERN TEST FAILURE,'<CRLF>
11117  060506  051523  046440  046505
11118  060514  051117  020131  047503
11119  060522  047125  020124  040520
11120  060530  052124  051105  020116
11121  060536  042524  052123  043040
11122  060544  044501  052514  042522
11123  060552  100054
11124  060554  047516  050040  051101          .ASCIZ  'NO PARITY ERROR OCCURS, BUT CAN NOT GET A HIT.'
11125  060562  052111  020131  051105
11126  060570  047522  020122  041517
11127  060576  052503  051522  020054
11128  060604  052502  020124  040503
11129  060612  020116  047516  020124
11130  060620  042507  020124  020101
11131  060626  044510  027124     000
11132
11133  060633     101  042104  042522  EM37:   .ASCIZ  'ADDRESS MEMORY COUNT PATTERN TEST, ERROR SUMMARY.'
11134  060640  051523  046440  046505
11135  060646  051117  020131  047503
11136  060654  047125  020124  040520
11137  060662  052124  051105  020116
11138  060670  042524  052123  020054
11139  060676  051105  047522  020122
11140  060704  052523  046515  051101
11141  060712  027131     000
11142
11143  060715     101  042104  042522  EM40:   .ASCII  'ADDRESS MEMORY COUNT PATTERN TEST FAILURE,'<CRLF>
11144  060722  051523  046440  046505
11145  060730  051117  020131  047503
11146  060736  047125  020124  040520
11147  060744  052124  051105  020116
11148  060752  042524  052123  043040
11149  060760  044501  052514  042522
11150  060766  100054
11151  060770  040503  044103  020105          .ASCII  'CACHE MEMORY ADDRESS PARITY ERROR OCCURRED'
11152  060776  042515  047515  054522
11153  061004  040440  042104  042522
11154  061012  051523  050040  051101
11155  061020  052111  020131  051105
11156  061026  047522  020122  041517
11157  061034  052503  051122  042105
11158  061042  040440  020124  044124          .ASCIZ  ' AT THE TEST ADDRESS.'
11159  061050  020105  042524  052123
11160  061056  040440  042104  042522
11161  061064  051523  000056
11162
11163  061070  042101  051104  051505  EM41:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILED TO GET '
11164  061076  020123  042515  047515
11165  061104  054522  042040  040525
11166  061112  020114  042101  051104
```

```
 11167  061120  051505  020123  042524
 11168  061126  052123  043040  044501
 11169  061134  042514  020104  047524
 11170  061142  043440  052105     040
 11171  061147     101  044040  052111          .ASCII  'A HIT AT A TEST ADDRESS,'<CRLF>
 11172  061154  040440  020124  020101
 11173  061162  042524  052123  040440
 11174  061170  042104  042522  051523
 11175  061176  100054
 11176  061200  044127  046111  020105          .ASCIZ  'WHILE WRITING THE ADDRESS MEMORY LOCATIONS.'
 11177  061206  051127  052111  047111
 11178  061214  020107  044124  020105
 11179  061222  042101  051104  051505
 11180  061230  020123  042515  047515
 11181  061236  054522  046040  041517
 11182  061244  052101  047511  051516
 11183  061252  000056
 11184
 11185  061254  042101  051104  051505  EM42:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILED TO GET'
 11186  061262  020123  042515  047515
 11187  061270  054522  042040  040525
 11188  061276  020114  042101  051104
 11189  061304  051505  020123  042524
 11190  061312  052123  043040  044501
 11191  061320  042514  020104  047524
 11192  061326  043440  052105
 11193  061332  020101  044510  020124          .ASCII  'A HIT AT A TEST ADDRESS,'<CRLF>
 11194  061340  052101  040440  052040
 11195  061346  051505  020124  042101
 11196  061354  051104  051505  026123
 11197  061362     200
 11198  061363     127  044510  042514          .ASCIZ  'WHILE READING BACK THE ADDRESS MEMORY LOCATIONS.'
 11199  061370  051040  040505  044504
 11200  061376  043516  041040  041501
 11201  061404  020113  044124  020105
 11202  061412  042101  051104  051505
 11203  061420  020123  042515  047515
 11204  061426  054522  046040  041517
 11205  061434  052101  047511  051516
 11206  061442  000056
 11207
 11208  061444  042101  051104  051505  EM43:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILURE,'<CRLF>
 11209  061452  020123  042515  047515
 11210  061460  054522  042040  040525
 11211  061466  020114  042101  051104
 11212  061474  051505  020123  042524
 11213  061502  052123  043040  044501
 11214  061510  052514  042522  100054
 11215  061516  040503  044103  020105          .ASCIZ  'CACHE ADDRESS MEMORY PARITY ERROR OCCURRED.'
 11216  061524  042101  051104  051505
 11217  061532  020123  042515  047515
 11218  061540  054522  050040  051101
 11219  061546  052111  020131  051105
 11220  061554  047522  020122  041517
 11221  061562  052503  051122  042105
 11222  061570  000056
```

```
11223
11224  061572  040515  047111  046440  EM44:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED,'
11225  061600  046505  051117  020131
11226  061606  054502  042524  046440
11227  061614  051501  020113  042507
11228  061622  042516  040522  047524
11229  061630  020122  042524  052123
11230  061636  043040  044501  042514
11231  061644  026104
11232  061646  042040  044517  043516          .ASCII  ' DOING CPU DATOB.'<CRLF>
11233  061654  041440  052520  042040
11234  061662  052101  041117  100056
11235  061670  020101  040515  047111          .ASCII  'A MAIN MEMORY ADDRESS AND CONTROL LINE '
11236  061676  046440  046505  051117
11237  061704  020131  042101  051104
11238  061712  051505  020123  047101
11239  061720  020104  047503  052116
11240  061726  047522  020114  044514
11241  061734  042516     040
11242  061737     120  051101  052111          .ASCIZ  'PARITY ERROR OCCURRED.'
11243  061744  020131  051105  047522
11244  061752  020122  041517  052503
11245  061760  051122  042105  000056
11246
11247  061766  040515  047111  046440  EM45:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED,'
11248  061774  046505  051117  020131
11249  062002  054502  042524  046440
11250  062010  051501  020113  042507
11251  062016  042516  040522  047524
11252  062024  020122  042524  052123
11253  062032  043040  044501  042514
11254  062040  026104
11255  062042  042040  044517  043516          .ASCII  ' DOING UNIBUS DATOB.'<CRLF>
11256  062050  052440  044516  052502
11257  062056  020123  040504  047524
11258  062064  027102     200
11259  062067     101  046440  044501          .ASCII  'A MAIN MEMORY ADDRESS AND CONTROL LINE '
11260  062074  020116  042515  047515
11261  062102  054522  040440  042104
11262  062110  042522  051523  040440
11263  062116  042116  041440  047117
11264  062124  051124  046117  046040
11265  062132  047111  020105
11266  062136  040520  044522  054524          .ASCIZ  'PARITY ERROR OCCURRED.'
11267  062144  042440  051122  051117
11268  062152  047440  041503  051125
11269  062160  042522  027104     000
11270
11271  062165     115  044501  020116  EM46:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED.'
11272  062172  042515  047515  054522
11273  062200  041040  052131  020105
11274  062206  040515  045523  043440
11275  062214  047105  051105  052101
11276  062222  051117  052040  051505
11277  062230  020124  040506  046111
11278  062236  042105     056
```

```
11279  062241     200  051127  047117          .ASCIZ  <CRLF>'WRONG BYTE WRITTEN, ON A CPU DATOB.'
11280  062246  020107  054502  042524
11281  062254  053440  044522  052124      '.
11282  062262  047105  020054  047117
11283  062270  040440  041440  052520
11284  062276  042040  052101  041117
11285  062304  000056
11286
11287  062306  040515  047111  046440  EM47:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED.'
11288  062314  046505  051117  020131
11289  062322  054502  042524  046440
11290  062330  051501  020113  042507
11291  062336  042516  040522  047524
11292  062344  020122  042524  052123
11293  062352  043040  044501  042514
11294  062360  027104
11295  062362  053600  047522  043516          .ASCIZ  <CRLF>'WRONG BYTE WRITTEN, ON A UNIBUS DATOB.'
11296  062370  041040  052131  020105
11297  062376  051127  052111  042524
11298  062404  026116  047440  020116
11299  062412  020101  047125  041111
11300  062420  051525  042040  052101
11301  062426  041117  000056
11302
11303          061572                  EM50=EM44
11304
11305          061766                  EM51=EM45
11306
11307          062165                  EM52=EM46
11308
11309          062306                  EM53=EM47
11310
11311  062432  040503  044103  020105  EM54:   .ASCII  'CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST FAILED.'
11312  062440  042101  051104  051505
11313  062446  020123  042515  047515
11314  062454  054522  050040  053517
11315  062462  051105  052440  020120
11316  062470  047111  040526  044514
11317  062476  040504  047524  020122
11318  062504  042524  052123  043040
11319  062512  044501  042514  027104
11320  062520  041600  041501  042510          .ASCII  <CRLF>'CACHE DATA OR ADDRESS MEMORY PARITY '
11321  062526  042040  052101  020101
11322  062534  051117  040440  042104
11323  062542  042522  051523  046440
11324  062550  046505  051117  020131
11325  062556  040520  044522  054524
11326  062564     040
11327  062565     105  051122  051117          .ASCIZ  'ERROR DETECTED.'
11328  062572  042040  052105  041505
11329  062600  042524  027104     000
11330
11331  062605     103  041501  042510  EM136:  .ASCII  'CACHE ADDRESS MEMORY PARITY LOGIC TEST FAILED.'<CRLF>
11332  062612  040440  042104  042522
11333  062620  051523  046440  046505
11334  062626  051117  020131  040520
```

```
11335  062634  044522  054524  046040
11336  062642  043517  041511  052040
11337  062650  051505  020124  040506
11338  062656  046111  042105  100056
11339  062664  047125  041101  042514          .ASCII  'UNABLE TO FORCE A PARITY ERROR ON THE LOW BYTE '
11340  062672  052040  020117  047506
11341  062700  041522  020105  020101
11342  062706  040520  044522  054524
11343  062714  042440  051122  051117
11344  062722  047440  020116  044124
11345  062730  020105  047514  020127
11346  062736  054502  042524     040
11347  062743     117  020106  047101          .ASCIZ  'OF AN ADDRESS.'<CRLF>'USING THE MAINTENANCE REGISTER.'
11348  062750  040440  042104  042522
11349  062756  051523  100054  051525
11350  062764  047111  020107  044124
11351  062772  020105  040515  047111
11352  063000  042524  040516  041516
11353  063006  020105  042522  044507
11354  063014  052123  051105  000056
11355
11356  063022  040503  044103  020105  EM137:  .ASCII  'CACHE ADDRESS MEMORY PARITY LOGIC TEST FAILED.'
11357  063030  042101  051104  051505
11358  063036  020123  042515  047515
11359  063044  054522  050040  051101
11360  063052  052111  020131  047514
11361  063060  044507  020103  042524
11362  063066  052123  043040  044501
11363  063074  042514  027104
11364  063100  052600  040516  046102          .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR ON THE HIGH BYTE '
11365  063106  020105  047524  043040
11366  063114  051117  042503  040440
11367  063122  050040  051101  052111
11368  063130  020131  051105  047522
11369  063136  020122  047117  052040
11370  063144  042510  044040  043511
11371  063152  020110  054502  042524
11372  063160     040
11373  063161     117  020106  047101          .ASCIZ  'OF AN ADDRESS,'<CRLF>'USING THE MAINTENANCE REGISTER.'
11374  063166  040440  042104  042522
11375  063174  051523  100054  051525
11376  063202  047111  020107  044124
11377  063210  020105  040515  047111
11378  063216  042524  040516  041516
11379  063224  020105  042522  044507
11380  063232  052123  051105  000056
11381
11382  063240                          EM140:
11383  063240  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
11384  063246  046505  051117  020131
11385  063254  040504  040524  050040
11386  063262  051101  052111  020131
11387  063270  044103  041505  042513
11388  063276  051522  052040  051505
11389  063304  020124  040506  046111
11390  063312  042105     056
```

```
11391  063315     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
11392  063322  042514  052040  020117
11393  063330  047506  041522  020105
11394  063336  020101  040520  044522
11395  063344  054524  042440  051122
11396  063352  051117  020054  051525
11397  063360  047111  020107
11398  063364  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11399  063372  047111  042524  040516
11400  063400  041516  020105  042522
11401  063406  044507  052123  051105
11402  063414  100054
11403  063416  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY EVEN WORD, LOW BYTE, PARITY '
11404  063424  046440  044501  020116
11405  063432  042515  047515  054522
11406  063440  042440  042526  020116
11407  063446  047527  042122  020054
11408  063454  047514  020127  054502
11409  063462  042524  020054  040520
11410  063470  044522  054524     040
11411  063475     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
11412  063502  051105  100054  051040
11413  063510  040505  044504  043516
11414  063516  040440  042040  052101
11415  063524  020101  040520  052124
11416  063532  051105  020116  044127
11417  063540  041511  020110
11418  063544  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
11419  063552  044040  053101  020105
11420  063560  040503  051525  042105
11421  063566  040440  020116  051105
11422  063574  047522  027122     000
11423
11424  063601                          EM141:
11425  063601     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
11426  063606  042515  047515  054522
11427  063614  042040  052101  020101
11428  063622  040520  044522  054524
11429  063630  041440  042510  045503
11430  063636  051105  020123  042524
11431  063644  052123  043040  044501
11432  063652  042514  027104
11433  063656  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
11434  063664  020105  047524  043040
11435  063672  051117  042503  040440
11436  063700  050040  051101  052111
11437  063706  020131  051105  047522
11438  063714  026122  052440  044523
11439  063722  043516     040
11440  063725     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11441  063732  044501  052116  047105
11442  063740  047101  042503  051040
11443  063746  043505  051511  042524
11444  063754  026122     200
11445  063757     101  020124  044124          .ASCII  'AT THE MAIN MEMORY ODD WORD, LOW BYTE, PARITY '
11446  063764  020105  040515  047111
```

```
11447  063772  046440  046505  051117
11448  064000  020131  042117  020104
11449  064006  047527  042122  020054
11450  064014  047514  020127  054502
11451  064022  042524  020054  040520
11452  064030  044522  054524     040
11453  064036     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
11454  064042  051105  100054  051040
11455  064050  040505  044504  043516
11456  064056  040440  042040  052101
11457  064064  020101  040520  052124
11458  064072  051105  020116  044127
11459  064100  041511  020110
11460  064104  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
11461  064112  044040  053101  020105
11462  064120  040503  051525  042105
11463  064126  040440  020116  051105
11464  064134  047522  027122     000
11465
11466  064141                          EM142:
11467  064141     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
11468  064146  042515  047515  054522
11469  064154  042040  052101  020101
11470  064162  040520  044522  054524
11471  064170  041440  042510  045503
11472  064176  051105  020123  042524
11473  064204  052123  043040  044501
11474  064212  042514  027104
11475  064216  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
11476  064224  020105  047524  043040
11477  064232  051117  042503  040440
11478  064240  050040  051101  052111
11479  064246  020131  051105  047522
11480  064254  026122  052440  044523
11481  064262  043516     040
11482  064265     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11483  064272  044501  052116  047105
11484  064300  047101  042503  051040
11485  064306  043505  051511  042524
11486  064314  026122     200
11487  064317     101  020124  044124          .ASCII  'AT THE MAIN MEMORY EVEN WORD, HIGH BYTE, PARITY '
11488  064324  020105  040515  047111
11489  064332  046440  046505  051117
11490  064340  020131  053105  047105
11491  064346  053440  051117  026104
11492  064354  044040  043511  020110
11493  064362  054502  042524  020054
11494  064370  040520  044522  054524
11495  064376     040
11496  064377     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
11497  064404  051105  100054  051040
11498  064412  040505  044504  043516
11499  064420  040440  042040  052101
11500  064426  020101  040520  052124
11501  064434  051105  020116  044127
11502  064442  041511  020110
```

```
11503  064446  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
11504  064454  044040  053101  020105
11505  064462  040503  051525  042105
11506  064470  040440  020116  051105
11507  064476  047522  027122     000
11508
11509  064503                          EM143:
11510  064503     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
11511  064510  042515  047515  054522
11512  064516  042040  052101  020101
11513  064524  040520  044522  054524
11514  064532  041440  042510  045503
11515  064540  051105  020123  042524
11516  064546  052123  043040  044501
11517  064554  042514  027104
11518  064560  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
11519  064566  020105  047524  043040
11520  064574  051117  042503  040440
11521  064602  050040  051101  052111
11522  064610  020131  051105  047522
11523  064616  026122  052440  044523
11524  064624  043516     040
11525  064627     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11526  064634  044501  052116  047105
11527  064642  047101  042503  051040
11528  064650  043505  051511  042524
11529  064656  026122     200
11530  064661     101  020124  044124          .ASCII  'AT THE MAIN MEMORY ODD WORD, HIGH BYTE, PARITY '
11531  064666  020105  040515  047111
11532  064674  046440  046505  051117
11533  064702  020131  042117  020104
11534  064710  047527  042122  020054
11535  064716  044510  044107  041040
11536  064724  052131  026105  050040
11537  064732  051101  052111  020131
11538  064740  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
11539  064746  026122  020200  042522
11540  064754  042101  047111  020107
11541  064762  020101  040504  040524
11542  064770  050040  052101  042524
11543  064776  047122  053440  044510
11544  065004  044103     040
11545  065007     123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
11546  065014  020104  040510  042526
11547  065022  041440  052501  042523
11548  065030  020104  047101  042440
11549  065036  051122  051117  000056
11550
11551  065044                          EM144:
11552  065044  040503  044103  020105          .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
11553  065052  040504  040524  046440
11554  065060  046505  051117  020131
11555  065066  040520  044522  054524
11556  065074  041440  042510  045503
11557  065102  051105  020123  042524
11558  065110  052123  043040  044501
```

```
11559  065116  042514  027104
11560  065122  052600  040516  046102          .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
11561  065130  020105  047524  043040
11562  065136  051117  042503  040440
11563  065144  050040  051101  052111
11564  065152  020131  051105  047522
11565  065160  026122  052440  044523
11566  065166  043516     040
11567  065171     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11568  065176  044501  052116  047105
11569  065204  047101  042503  051040
11570  065212  043505  051511  042524
11571  065220  026122     200
11572  065223     101  020124  044124          .ASCII  'AT THE GROUP ZERO,LOW BYTE, DATA PARITY CHECKER,'
11573  065230  020105  051107  052517
11574  065236  020120  042532  047522
11575  065244  046054  053517  041040
11576  065252  052131  026105  042040
11577  065260  052101  020101  040520
11578  065266  044522  054524  041440
11579  065274  042510  045503  051105
11580  065302     054
11581  065303     200  042522  042101          .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
11582  065310  047111  020107  020101
11583  065316  040504  040524  050040
11584  065324  052101  042524  047122
11585  065332  053440  044510  044103
11586  065340  051440  047510  046125
11587  065346  020104  040510  042526
11588  065354     040
11589  065355     103  052501  042523          .ASCIZ  'CAUSED AN ERROR.'
11590  065362  020104  047101  042440
11591  065370  051122  051117  000056
11592
11593  065376                                  EM145:
11594  065376  040503  044103  020105          .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
11595  065404  040504  040524  046440
11596  065412  046505  051117  020131
11597  065420  040520  044522  054524
11598  065426  041440  042510  045503
11599  065434  051105  020123  042524
11600  065442  052123  043040  044501
11601  065450  042514  027104
11602  065454  052600  040516  046102          .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
11603  065462  020105  047524  043040
11604  065470  051117  042503  040440
11605  065476  050040  051101  052111
11606  065504  020131  051105  047522
11607  065512  026122  052440  044523
11608  065520  043516     040
11609  065523     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11610  065530  044501  052116  047105
11611  065536  047101  042503  051040
11612  065544  043505  051511  042524
11613  065552  026122     200
11614  065555     101  020124  044124          .ASCII  'AT THE GROUP ONE,LOW BYTE, DATA PARITY CHECKER,'
```

```
11615  065562  020105  051107  052517
11616  065570  020120  047117  026105
11617  065576  047514  020127  054502
11618  065604  042524  020054  040504
11619  065612  040524  050040  051101
11620  065620  052111  020131  044103
11621  065626  041505  042513  026122
11622  065634  051200  040505  044504          .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
11623  065642  043516  040440  042040
11624  065650  052101  020101  040520
11625  065656  052124  051105  020116
11626  065664  044127  041511  020110
11627  065672  044123  052517  042114
11628  065700  044040  053101  020105
11629  065706  040503  051525  042105          .ASCIZ  'CAUSED AN ERROR.'
11630  065714  040440  020116  051105
11631  065722  047522  027122     000
11632
11633  065727                          EM146:
11634  065727     103  041501  042510          .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
11635  065734  042040  052101  020101
11636  065742  042515  047515  054522
11637  065750  050040  051101  052111
11638  065756  020131  044103  041505
11639  065764  042513  051522  052040
11640  065772  051505  020124  040506
11641  066000  046111  042105     056
11642  066005     200  047125  041101          .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
11643  066012  042514  052040  020117
11644  066020  047506  041522  020105
11645  066026  020101  040520  044522
11646  066034  054524  042440  051122
11647  066042  051117  020054  051525
11648  066050  047111  020107
11649  066054  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11650  066062  047111  042524  040516
11651  066070  041516  020105  042522
11652  066076  044507  052123  051105
11653  066104  100054
11654  066106  052101  052040  042510          .ASCII  'AT THE GROUP ZERO,HIGH BYTE, DATA PARITY CHECKER,'
11655  066114  043440  047522  050125
11656  066122  055040  051105  026117
11657  066130  044510  044107  041040
11658  066136  052131  026105  042040
11659  066144  052101  020101  040520
11660  066152  044522  054524  041440
11661  066160  042510  045503  051105
11662  066166     054
11663  066167     200  042522  042101          .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
11664  066174  047111  020107  020101
11665  066202  040504  040524  050040
11666  066210  052101  042524  047122
11667  066216  053440  044510  044103
11668  066224  051440  047510  046125
11669  066232  020104  040510  042526
11670  066240     040
```

```
11671  066241     103  052501  042523          .ASCIZ  'CAUSED AN ERROR.'
11672  066246  020104  047101  042440
11673  066254  051122  051117  000056
11674
11675  066262                          EM147:
11676  066262  040503  044103  020105          .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAI_ED.'
11677  066270  040504  040524  046440
11678  066276  046505  051117  020131
11679  066304  040520  044522  054524
11680  066312  041440  042510  045503
11681  066320  051105  020123  042524
11682  066326  052123  043040  044501
11683  066334  042514  027104
11684  066340  052600  040516  046102          .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
11685  066346  020105  047524  043040
11686  066354  051117  042503  040440
11687  066362  050040  051101  052111
11688  066370  020131  051105  047522
11689  066376  026122  052440  044523
11690  066404  043516     040
11691  066407     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
11692  066414  044501  052116  047105
11693  066422  047101  042503  051040
11694  066430  043505  051511  042524
11695  066436  026122     200
11696  066441     101  020124  044124          .ASCII  'AT THE GROUP ONE,HIGH BYTE, DATA PARITY CHECKER,'
11697  066446  020105  051107  052517
11698  066454  020120  047117  026105
11699  066462  044510  044107  041040
11700  066470  052131  026105  042040
11701  066476  052101  020101  040520
11702  066504  044522  054524  041440
11703  066512  042510  045503  051105
11704  066520     054
11705  066521     200  042522  042101          .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
11706  066526  047111  020107  020101
11707  066534  040504  040524  050040
11708  066542  052101  042524  047122
11709  066550  053440  044510  044103
11710  066556  051440  047510  046125
11711  066564  020104  040510  042526
11712  066572     040
11713  066573     103  052501  042523          .ASCIZ  'CAUSED AN ERROR.'
11714  066600  020104  047101  042440
11715  066606  051122  051117  000056
11716
11717  066614  052600  042516  050130  EM150:  .ASCIZ  <CRLF>'UNEXPECTED CPU ERROR TRAPPED TO VECTOR ERRVEC (4)!'
11718  066622  041505  042524  020104
11719  066630  050103  020125  051105
11720  066636  047522  020122  051124
11721  066644  050101  042520  020104
11722  066652  047524  053040  041505
11723  066660  047524  020122  051105
11724  066666  053122  041505  024040
11725  066674  024464  000041
11726
```

```
11727  066700  040515  051523  041040  EM151:  .ASCIZ  'MASS BUS WRITE HIT DID NOT INVALIDATE THE CACHE.'
11728  066706  051525  053440  044522
11729  066714  042524  044040  052111
11730  066722  042040  042111  047040
11731  066730  052117  044440  053116
11732  066736  046101  042111  052101
11733  066744  020105  044124  020105
11734  066752  040503  044103  027105
11735  066760     000
11736
11737          066700                  EM152=EM151
11738          066700                  EM153=EM151
11739
11740  066761     104  053105  041511  EM154:  .ASCIZ  'DEVICE ERROR IN THE RS04.'
11741  066766  020105  051105  047522
11742  066774  020122  047111  052040
11743  067002  042510  051040  030123
11744  067010  027064     000
11745
11746  067013     104  053105  041511  EM155:  .ASCIZ  'DEVICE ERROR IN THE RP04.'
11747  067020  020105  051105  047522
11748  067026  020122  047111  052040
11749  067034  042510  051040  030120
11750  067042  027064     000
11751
11752  067045     104  053105  041511  EM156:  .ASCIZ  'DEVICE ERROR IN THE MASS BUS TESTER.'
11753  067052  020105  051105  047522
11754  067060  020122  047111  052040
11755  067066  042510  046440  051501
11756  067074  020123  052502  020123
11757  067102  042524  052123  051105
11758  067110  000056
11759
11760
11761  067112  042504  044526  042503  EM160:  .ASCIZ  'DEVICE ERROR IN THE RK05.'
11762  067120  042440  051122  051117
11763  067126  044440  020116  044124
11764  067134  020105  045522  032460
11765  067142  000056
11766
11767  067144  042504  044526  042503  EM161:  .ASCIZ  'DEVICE ERROR IN THE UNIBUS EXECIZER.'
11768  067152  042440  051122  051117
11769  067160  044440  020116  044124
11770  067166  020105  047125  041111
11771  067174  051525  042440  042530
11772  067202  044503  042532  027122
11773  067210     000
11774
11775                          ;THESE ARE DATA HEADERS:
11776
11777  067211     040  052040  051505  DH1:    .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'PHYSICAL ADDR.'<TAB>'CALL AT PC.'
11778  067216  027124  020011  051107
11779  067224  052517  027120  050011
11780  067232  054510  044523  040503
11781  067240  020114  042101  051104
11782  067246  004456  040503  046114
```

```
11783  067254  040440  020124  041520
11784  067262  000056
11785
11786  067264  020040  042524  052123  DH2:    .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR ADDR REG.'<TAB>'ERROR REG.'<TAB>
11787  067272  004456  043440  047522
11788  067300  050125  004456  051105
11789  067306  047522  020122  042101
11790  067314  051104  051040  043505
11791  067322  004456  051105  047522
11792  067330  020122  042522  027107
11793  067336     011
11794  067337     122  043105  040440          .ASCIZ  'REF ADDR.'<TAB>'TRAP AT PC.'
11795  067344  042104  027122  052011
11796  067352  040522  020120  052101
11797  067360  050040  027103     000
11798
11799          067264                  DH3=DH2
11800
11801          067264                  DH4=DH2
11802
11803  067365     040  052040  051505  DH5:    .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'READ.'
11804  067372  027124  041411  046101
11805  067400  020114  052101  050040
11806  067406  027103  051011  040505
11807  067414  027104     000
11808
11809          067365                  DH6=DH5
11810
11811          067365                  DH7=DH5
11812
11813          067365                  DH10=DH5
11814
11815  067417     040  052040  051505  DH11:   .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'TRAP AT PC.'<TAB>'ERROR ADDR REG.'
11816  067424  027124  020011  051107
11817  067432  052517  027120  052011
11818  067440  040522  020120  052101
11819  067446  050040  027103  042411
11820  067454  051122  051117  040440
11821  067462  042104  020122  042522
11822  067470  027107     000
11823
11824  067473     040  052040  051505  DH12:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'CALL AT PC.'<TAB>'TEST ADDR.'<TAB>
11825  067500  027124  020011  051107
11826  067506  052517  027120  041411
11827  067514  046101  020114  052101
11828  067522  050040  027103  052011
11829  067530  051505  020124  042101
11830  067536  051104  004456
11831  067542  040504  040524  053440          .ASCIZ  'DATA WR. DATA READ.'
11832  067550  027122  042040  052101
11833  067556  020101  042522  042101
11834  067564  000056
11835
11836  067566  020040  042524  052123  DH13:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'*DATA.'<TAB>'+DATA.'<TAB>
11837  067574  004456  043440  047522
11838  067602  050125  004456  042052
```

```
11839  067610  052101  027101  025411
11840  067616  040504  040524  004456
11841  067624  051105  047522  020122          .ASCIZ  'ERROR COUNT.'
11842  067632  047503  047125  027124
11843  067640     000
11844
11845  067641     040  052040  051505  DH14:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADDR REG.'
11846  067646  027124  041411  046101
11847  067654  020114  052101  050040
11848  067662  027103  042411  051122
11849  067670  051117  040440  042104
11850  067676  020122  042522  027107
11851  067704  052011  040522  020120          .ASCII  <TAB>'TRAP AT PC.'<TAB>
11852  067712  052101  050040  027103
11853  067720     011
11854  067721     105  051122  051117          .ASCIZ  'ERROR REG.'
11855  067726  051040  043505  000056
11856
11857  067734  020040  042524  052123  DH15:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
11858  067742  004456  040503  046114
11859  067750  040440  020124  041520
11860  067756  000056
11861
11862  067760  020040  042524  052123  DH16:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'WROTE.'<TAB>'READ.'<TAB>
11863  067766  004456  043440  047522
11864  067774  050125  004456  051127
11865  070002  052117  027105  051011
11866  070010  040505  027104     011
11867  070015     101  042104  020122          .ASCIZ  'ADDR TESTED.'<TAB>'CALL AT PC.'
11868  070022  042524  052123  042105
11869  070030  004456  040503  046114
11870  070036  040440  020124  041520
11871  070044  000056
11872
11873  070046  020040  042524  052123  DH17:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR AT PC.'<TAB>'READ.'<TAB>
11874  070054  004456  043440  047522
11875  070062  050125  004456  051105
11876  070070  047522  020122  052101
11877  070076  050040  027103  051011
11878  070104  040505  027104     011
11879  070111     111  027116  040411          .ASCIZ  'IN.'<TAB>'ADDRESS.'
11880  070116  042104  042522  051523
11881  070124  000056
11882
11883          070046                  DH20=DH17
11884
11885  070126  020040  042524  052123  DH21:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'READ.'<TAB>' GROUP.'<TAB>'ADDRESS.'
11886  070134  004456  040503  046114
11887  070142  040440  020124  041520
11888  070150  004456  042522  042101
11889  070156  004456  043440  047522
11890  070164  050125  004456  042101
11891  070172  051104  051505  027123
11892  070200     000
11893
11894  070201     040  052040  051505  DH22:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'EXPECTED ERROR AT.'
```

```
11895  070206  027124  041411  046101
11896  070214  020114  052101  050040
11897  070222  027103  042411  050130
11898  070230  041505  042524  020104
11899  070236  051105  047522  020122
11900  070244  052101  000056
11901
11902  070250  020040  042524  052123  DH23:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'EXPECTED ADRS.'<TAB>
11903  070256  004456  040503  046114
11904  070264  040440  020124  041520
11905  070272  004456  054105  042520
11906  070300  052103  042105  040440
11907  070306  051104  027123     011
11908  070313     107  052117  040440          .ASCIZ  'GOT ADRS.'<TAB>'ERROR REG.'
11909  070320  051104  027123  042411
11910  070326  051122  051117  051040
11911  070334  043505  000056
11912
11913          070201                     DH24=DH22
11914
11915          070250                     DH25=DH23
11916
11917  070340  020040  042524  052123  DH26:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
11918  070346  004456  040503  046114
11919  070354  040440  020124  041520
11920  070362  004456  043440  047522
11921  070370  050125  004456  042101
11922  070376  051104  051505  027123
11923  070404     000
11924
11925  070405     040  052040  051505  DH27:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ESTABLISHED HIT.'
11926  070412  027124  041411  046101
11927  070420  020114  052101  050040
11928  070426  027103  020011  051107
11929  070434  052517  027120  042411
11930  070442  052123  041101  044514
11931  070450  044123  042105  044040
11932  070456  052111     056
11933  070461     040  052502  020124          .ASCIZ  ' BUT GOT HIT.'
11934  070466  047507  020124  044510
11935  070474  027124     000
11936
11937          070201                     DH30=DH22
11938
11939          070250                     DH31=DH23
11940
11941          070201                     DH32=DH22
11942
11943          070250                     DH33=DH23
11944
11945  070477     040  052040  051505  DH34:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'READ.'<TAB>'IN ADDRESS.'<TAB>
11946  070504  027124  050011  020103
11947  070512  043117  041440  046101
11948  070520  027114  051011  040505
11949  070526  027104  044411  020116
11950  070534  042101  051104  051505
```

.

.

```
11951  070542  027123     011
11952  070545     105  050130  041505          .ASCIZ  'EXPECTED.'
11953  070552  042524  027104     000
11954
11955          070477                  DH35=DH34
11956
11957  070557     040  052040  051505  DH36:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
11958  070564  027124  041411  046101
11959  070572  020114  052101  050040
11960  070600  027103  020011  051107
11961  070606  052517  027120  040411
11962  070614  042104  042522  051523
11963  070622  000056
11964
11965  070624  020040  042524  052123  DH37:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR COUNT.'<TAB>
11966  070632  004456  043440  047522
11967  070640  050125  004456  051105
11968  070646  047522  020122  047503
11969  070654  047125  027124     011
11970  070661     052  041040  042101          .ASCIZ  '* BAD ADRS.'<TAB>'+ BAD ADRS.'
11971  070666  040440  051104  027123
11972  070674  025411  041040  042101
11973  070702  040440  051104  027123
11974  070710     000
11975
11976
11977  070711     040  052040  051505  DH41:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
11978  070716  027124  041411  046101
11979  070724  020114  052101  050040
11980  070732  027103  020011  051107
11981  070740  052517  027120  040411
11982  070746  042104  042522  051523
11983  070754  000056
11984
11985          070711                  DH42=DH41
11986
11987  070756  020040  042524  052123  DH43:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>' GROUP.'
11988  070764  004456  040503  046114
11989  070772  040440  020124  041520
11990  071000  004456  051124  050101
11991  071006  040440  020124  041520
11992  071014  004456  043440  047522
11993  071022  050125     056
11994
11995          070756                  DH40=DH43
11996
11997  071025     040  052040  051505  DH44:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
11998  071032  027124  041411  046101
11999  071040  020114  052101  050040
12000  071046  027103  052011  040522
12001  071054  020120  052101  050040
12002  071062  027103     011
12003  071065     105  051122  051117          .ASCIZ  'ERROR ADRS REG.'<TAB>'ERROR REG.'
12004  071072  040440  051104  020123
12005  071100  042522  027107  042411
12006  071106  051122  051117  051040
```

```
12007  071114  043505  000056
12008
12009          071025                  DH45=DH44
12010
12011  071120  020040  042524  052123  DH46:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
12012  071126  004456  040503  046114
12013  071134  040440  020124  041520
12014  071142  000056
12015
12016          071120                  DH47=DH46
12017
12018          071025                  DH50=DH44
12019
12020          071025                  DH51=DH44
12021
12022          071120                  DH52=DH46
12023
12024          071120                  DH53=DH46
12025
12026  071144  020040  042524  052123  DH54:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR COUNT.'
12027  071152  004456  040503  046114
12028  071160  040440  020124  041520
12029  071166  004456  051105  047522
12030  071174  020122  047503  047125
12031  071202  027124     000
12032
12033  071205     040  052040  051505  DH136:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
12034  071212  027124  041411  046101
12035  071220  020114  052101  050040
12036  071226  027103  020011  051107
12037  071234  052517  027120  040411
12038  071242  042104  042522  051523
12039  071250  000056
12040
12041          071205                  DH137=DH136
12042
12043  071252  020040  042524  052123  DH140:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'DATA.'<TAB>'ADDRESS.'
12044  071260  004456  040503  046114
12045  071266  040440  020124  041520
12046  071274  004456  040504  040524
12047  071302  004456  042101  051104
12048  071310  051505  027123     000
12049
12050          071252                  DH141=DH140
12051
12052          071252                  DH142=DH140
12053
12054          071252                  DH143=DH140
12055
12056          071252                  DH144=DH140
12057
12058          071252                  DH145=DH140
12059
12060          071252                  DH146=DH140
12061
12062          071252                  DH147=DH140
```

```
12063
12064  071315  000040  052040  051505  DH150:  .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'CPU ERROR REGISTER.'
12065  071322  027124  052011  040522
12066  071330  020120  052101  050040
12067  071336  027103  041411  046101
12068  071344  020114  052101  050040
12069  071352  027103  041411  052520
12070  071360  042440  051122  051117
12071  071366  051040  043505  051511
12072  071374  042524  027122  000000
12073
12074  071401  000125  044523  043516  DH151:  .ASCII  'USING THE RS04.'
12075  071406  052040  042510  051040
12076  071414  030123  027064
12077  071420  020040  042524  052123          .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
12078  071426  004456  051107  052517
12079  071434  027120  040411  042104
12080  071442  042522  051523  000056
12081
12082  071450  051525  047111  020107  DH152:  .ASCII  'USING THE RP04.'
12083  071456  044124  020105  050122
12084  071464  032060  000056
12085  071467  000040  052040  051505          .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
12086  071474  027124  043411  047522
12087  071502  050125  004456  042101
12088  071510  051104  051505  027123
12089  071516  000000
12090
12091  071517  000125  044523  043516  DH153:  .ASCII  'USING THE MASS BUS TESTER.'
12092  071524  052040  042510  046440
12093  071532  051501  020123  052502
12094  071540  020123  042524  052123
12095  071546  051105  000056
12096  071551  000040  052040  051505          .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
12097  071556  027124  043411  047522
12098  071564  050125  004456  042101
12099  071572  051104  051505  027123
12100  071600  000000
12101
12102  071601  000040  052040  051505  DH154:  .ASCIZ  '  TEST.'<TAB>'RS4CS2.'<TAB>'RS4DS.'<TAB>'RS4ER.'
12103  071606  027124  051011  032123
12104  071614  051503  027062  051011
12105  071622  032123  051504  004456
12106  071630  051522  042464  027122
12107  071636  000000
12108
12109  071637  000040  052040  051505  DH155:  .ASCIZ  '  TEST.'<TAB>'RP4CS2.'<TAB>'RP4DS.'<TAB>'RP4ER.'
12110  071644  027124  051011  032120
12111  071652  051503  027062  051011
12112  071660  032120  051504  004456
12113  071666  050122  042464  027122
12114  071674  000000
12115
12116  071675  000040  052040  051505  DH156:  .ASCIZ  '  TEST.'<TAB>'RH4CS2.'<TAB>'RH4ST.'<TAB>'RH4ER.'
12117  071702  027124  051011  032110
12118  071710  051503  027062  051011
```

```
12119  071716  032110  052123  004456
12120  071724  044122  042464  027122
12121  071732     000
12122
12123
12124  071733     040  052040  051505  DH160:  .ASCIZ  '  TEST.'<TAB>'RK5ER.'<TAB>'RK5DS.'
12125  071740  027124  051011  032513
12126  071746  051105  004456  045522
12127  071754  042065  027123     000
12128
12129  071761     040  052040  051505  DH161:  .ASCIZ  '  TEST.'<TAB>'UBECR1.'<TAB>'UBECR2.'
12130  071766  027124  052411  042502
12131  071774  051103  027061  052411
12132  072002  042502  051103  027062
12133  072010     000
12134
12135                              ;THESE ARE DATA FORMAT DESIGNATORS FOR THE DATA TABLE:
12136
12137  072011     004     004     003  DF1:    .BYTE   4,4,3,3
12138  072014     003
12139
12140  072015     004     004     007  DF2:    .BYTE   4,4,7,0,3,3
12141  072020     000     003     003
12142
12143          072015                  DF3=DF2
12144
12145          072015                  DF4=DF2
12146
12147  072023     004     003     000  DF5:    .BYTE   4,3,0,5,0,0,0,0
12148  072026     005     000     000
12149  072031     000     000
12150
12151          072023                  DF6=DF5
12152
12153          072023                  DF7=DF5
12154
12155          072023                  DF10=DF5
12156
12157  072033     004     004     003  DF11:   .BYTE   4,4,3,7,5,0,5,3,0
12158  072036     007     005     000
12159  072041     005     003     000
12160
12161  072044     004     004     003  DF12:   .BYTE   4,4,3,3,0,0
12162  072047     003     000     000
12163
12164  072052     004     004     000  DF13:   .BYTE   4,4,0,0,4
12165  072055     000     004
12166
12167  072057     004     003     007  DF14:   .BYTE   4,3,7,3,0
12168  072062     003     000
12169
12170  072064     004     003          DF15:   .BYTE   4,3
12171
12172  072066     004     004     000  DF16:   .BYTE   4,4,0,0,3,3
12173  072071     000     003     003
12174
```

```
12175  072074    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
12176  072077    000    005    003
12177  072102    005    005    005
12178  072105    003    005    003
12179  072110    005    003    005
12180  072113    003    005    000
12181  072116    005    000    005
12182  072121    000    005    000
12183
12184          072074                DF20=DF17
12185
12186  072124    004    003    000  DF21:   .BYTE   4,3,0,4,3,5
12187  072127    004    003    005
12188  072132    005    003    005          .BYTE   5,3,5,3,5,3,5,3,5
12189  072136    003    005    003
12190  072140    005    003    005
12191  072143    000    005    000          .BYTE   0,5,0,5,0,5,0
12192  072146    005    000    005
12193  072151    000
12194
12195  072152    004    003    002  DF22:   .BYTE   4,3,2
12196
12197  072155    004    003    002  DF23:   .BYTE   4,3,2,2,0
12198  072160    002    000
12199
12200          072152                DF24=DF22
12201
12202          072155                DF25=DF23
12203
12204  072162    004    003    004  DF26:   .BYTE   4,3,4,2
12205  072165    002
12206
12207  072166    004    003    004  DF27:   .BYTE   4,3,4,2,2
12208  072171    002    002
12209
12210          072152                DF30=DF22
12211
12212          072155                DF31=DF23
12213
12214          072152                DF32=DF22
12215
12216          072155                DF33=DF23
12217
12218  072173    004    003    000  DF34:   .BYTE   4,3,0,2,0
12219  072176    002    000
12220
12221          072173                DF35=DF34
12222
12223  072200    004    003    004  DF36:   .BYTE   4,3,4,2
12224  072203    002
12225
12226  072204    004    004    007  DF37:   .BYTE   4,4,7,2,2,0
12227  072207    002    002    000
12228
12229
12230  072212    004    003    004  DF41:   .BYTE   4,3,4,2
```

```
12231  072215     002
12232
12233          072212                    DF42=DF41
12234
12235  072216     004     003     003    DF43:   .BYTE   4,3,3,4,5,2,7,0
12236  072221     004     005     002
12237  072224     007     000
12238
12239          072216                    DF40=DF43
12240
12241  072226     004     003     002    DF44:   .BYTE   4,3,2,7,0,5,2,5,0,5,2,5,0,5,2
12242  072231     007     000     005
12243  072234     002     005     000
12244  072237     005     002     005
12245  072242     000     005     002
12246
12247          072226                    DF45=DF44
12248
12249  072245     004     003     005    DF46:   .BYTE   4,3,5,2,5,0,5,2,5,0,5,2
12250  072250     002     005     000
12251  072253     005     002     005
12252  072256     000     005     002
12253
12254          072245                    DF47=DF46
12255
12256          072226                    DF50=DF44
12257
12258          072226                    DF51=DF44
12259
12260          072245                    DF52=DF46
12261
12262          072245                    DF53=DF46
12263
12264  072261     004     003     004    DF54:   .BYTE   4,3,4
12265
12266  072264     004     003     004    DF136:  .BYTE   4,3,4,2
12267  072267     002
12268
12269          072264                    DF137=DF136
12270
12271  072270     004     003     000    DF140:  .BYTE   4,3,0,2
12272  072273     002
12273
12274          072270                    DF141=DF140
12275
12276          072270                    DF142=DF140
12277
12278          072270                    DF143=DF140
12279
12280          072270                    DF144=DF140
12281
12282          072270                    DF145=DF140
12283
12284          072270                    DF146=DF140
12285
12286          072270                    DF147=DF140
```

```
12287
12288  072274  004     003     003  DF150:  .BYTE   4,3,3,0
12289  072277  000
12290
12291  072300  004     004     007  DF151:  .BYTE   4,4,7
12292
12293          072300                DF152=DF151
12294          072300                DF153=DF151
12295
12296  072303  004     000     000  DF154:  .BYTE   4,0,0,0
12297  072306  000
12298
12299          072303                DF155=DF154
12300          072303                DF156=DF154
12301          072303                DF157=DF154
12302          072303                DF160=DF154
12303          072303                DF161=DF154
12304
12305
12306          072310                        .EVEN
12307
12308                                ;THESE ARE DATA TABLES:
12309
12310  072310  001632  001634  001636  DT1:    .WORD   $TMP0,$TMP1,$TMP2,$ERRPC,0
12311  072316  001516  000000
12312
12313  072322  001632  001646  001636  DT2:    .WORD   $TMP0,$TMP6,$TMP2,$TMP1,$TMP5,$TMP4,0
12314  072330  001634  001644  001642
12315  072336  000000
12316
12317          072322                DT3=DT2
12318
12319          072322                DT4=DT2
12320
12321  072340  001632  001516  001636  DT5:    .WORD   $TMP0,$ERRPC,$TMP2,MTA5,JJPAT1,JJPAT2,JJPAT3,JJPAT4,0
12322  072346  052462  023770  023772
12323  072354  023774  023776  000000
12324
12325          072340                DT6=DT5
12326
12327          072340                DT7=DT5
12328
12329          072340                DT10=DT5
12330
12331  072362  001632  001634  001636  DT11:   .WORD   $TMP0,$TMP1,$TMP2,$TMP4,MTA11,$TMP3,$TAB,$TMP7,$TMP6,0
12332  072370  001642  052544  001640
12333  072376  052460  001650  001646
12334  072404  000000
12335
12336  072406  001632  001634  001516  DT12:   .WORD   $TMP0,$TMP1,$ERRPC,$TMP3,$TMP4,$TMP5,0
12337  072414  001640  001642  001644
12338  072422  000000
12339
12340  072424  001632  001634  001636  DT13:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,0
12341  072432  001640  001642  000000
12342
```

```
12343  072440  001632  001516  001634  DT14:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,0
12344  072446  001640  001642  000000
12345
12346  072454  001632  001634  000000  DT15:   .WORD   $TMP0,$TMP1,0
12347
12348  072462  001632  001634  001636  DT16:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,$ERRPC,0
12349  072470  001640  001642  001516
12350  072476  000000
12351
12352  072500  001632  001634  001636  DT17:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,MTC17,$TMP4,$CRLF,MTB17
12353  072506  001640  052636  001642
12354  072514  001713  052616
12355  072520  052611  001644  052611          .WORD   MTA17,$TMP5,MTA17,$TMP6,MTA17,$TMP7,MTA17,$TMP10
12356  072526  001646  052611  001650
12357  072534  052611  001652
12358  072540  001713  033236  052460          .WORD   $CRLF,MMPAT1,$TAB,MMPAT2,$TAB,MMPAT3,$TAB,MMPAT4,0
12359  072546  033240  052460  033242
12360  072554  052460  033244  000000
12361
12362  072562  001632  001634  001636  DT20:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,MTA20,$TMP4,$CRLF,MTB17
12363  072570  001640  052645  001642
12364  072576  001713  052616
12365  072602  001644  052611  001646          .WORD   $TMP5,MTA17,$TMP6,MTA17,$TMP7,MTA17,$TMP10,MTA17
12366  072610  052611  001650  052611
12367  072616  001652  052611
12368  072622  001713  033236  052460          .WORD   $CRLF,MMPAT1,$TAB,MMPAT3,$TAB,MMPAT3,$TAB,MMPAT4,0
12369  072630  033242  052460  033242
12370  072636  052460  033244  000000
12371
12372  072644  001632  001634  001636  DT21:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,MTA21
12373  072652  001640  001642  052654
12374  072660  052611  001644  052611          .WORD   MTB21,$TMP5,MTB21,$TMP6,MTB21,$TMP7,MTB21,$TMP10,$CRLF
12375  072666  001646  052611  001650
12376  072674  052611  001652  001713
12377  072702  031372  052460  031374          .WORD   KKPAT1,$TAB,KKPAT2,$TAB,KKPAT3,$TAB,KKPAT4,0
12378  072710  052460  031376  052460
12379  072716  031400  000000
12380
12381  072722  001632  001516  005324  DT22:   .WORD   $TMP0,$ERRPC,XADR2,0
12382  072730  000000
12383
12384  072732  001632  001516  005324  DT23:   .WORD   $TMP0,$ERRPC,XADR2,$TMP3,$TMP1,0
12385  072740  001640  001634  000000
12386
12387  072746  001632  001516  006224  DT24:   .WORD   $TMP0,$ERRPC,XXADR2,0
12388  072754  000000
12389
12390  072756  001632  001516  006224  DT25:   .WORD   $TMP0,$ERRPC,XXADR2,$TMP3,$TMP1,0
12391  072764  001640  001634  000000
12392
12393  072772  001632  001516  001634  DT26:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP2,0
12394  073000  001636  000000
12395
12396  073004  001632  001516  001634  DT27:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP2,$TMP4,0
12397  073012  001636  001642  000000
12398
```

```
12399  073020  001632  001516  007142  DT30:   .WORD   $TMP0,$ERRPC,RRADR2,0
12400  073026  000000
12401
12402  073030  001632  001516  007142  DT31:   .WORD   $TMP0,$ERRPC.RRADR2,$TMP3,$TMP1,0
12403  073036  001640  001634  000000
12404
12405  073044  001632  001516  010024  DT32:   .WORD   $TMP0,$ERRPC,SSADR2,0
12406  073052  000000
12407
12408  073054  001632  001516  010024  DT33:   .WORD   $TMP0,$ERRPC,SSADR2,$TMP3,$TMP1,0
12409  073062  001640  001634  000000
12410
12411  073070  001632  001516  001636  DT34:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,$TMP5,0
12412  073076  001640  001644  000000
12413
12414          073070                  DT35=DT34
12415
12416  073104  001632  001516  001636  DT36:   .WORD   $TMP0,$ERRPC,$TMP2,BBADR1,0
12417  073112  012630  000000
12418
12419  073116  001632  001634  012650  DT37:   .WORD   $TMP0,$TMP1,BBCNT1,BBADR2,BBADR3,0
12420  073124  012634  012640  000000
12421
12422
12423  073132  001632  001516  001636  DT41:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
12424  073140  001640  000000
12425
12426          073132                  DT42=DT41
12427
12428  073144  001632  001516  001636  DT43:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA43,$TMP5,$TMP7,$TMP4,0
12429  073152  001640  052741  001644
12430  073160  001650  001642  000000
12431
12432          073144                  DT40=DT43
12433
12434  073166  001632  001516  001666  DT44:   .WORD   $TMP0,$ERRPC,$TMP16,$TMP3,$TMP5,MTA45,$TMP12,MTB45
12435  073174  001640  001644  053014
12436  073202  001656  053042
12437  073206  001652  053057  001646          .WORD   $TMP10,MTC45,$TMP6,MTB45,$TMP11,MTC45,$TMP14,0
12438  073214  053042  001654  053057
12439  073222  001662  000000
12440
12441          073166                  DT45=DT44
12442
12443  073226  001632  001656  053014  DT46:   .WORD   $TMP0,$TMP12,MTA45,$TMP10,MTB45,$TMP6,MTC45
12444  073234  001652  053042  001646
12445  073242  053057
12446  073244  001636  053042  001650          .WORD   $TMP2,MTB45,$TMP7,MTC45,$TMP4,0
12447  073252  053057  001642  000000
12448
12449          073226                  DT47=DT46
12450
12451  073260  001632  001516  001666  DT50:   .WORD   $TMP0,$ERRPC,$TMP16,$TMP3,$TMP5,MTA50,$TMP12,MTB45
12452  073266  001640  001644  053072
12453  073274  001656  053042
12454  073300  001652  053057  001646          .WORD   $TMP10,MTC45,$TMP6,MTB45,$TMP11,MTC45,$TMP14,0
```

```
12455  073306  053042  001654  053057
12456  073314  001662  000000
12457
12458          073260                  DT51=DT50
12459
12460  073320  001632  001656  053072  DT52:   .WORD   $TMP0,$TMP12,MTA50,$TMP10,MTB45,$TMP6,MTC45
12461  073326  001652  053042  001646
12462  073334  053057
12463  073336  001636  053042  001650          .WORD   $TMP2,MTB45,$TMP7,MTC45,$TMP4,0
12464  073344  053057  001642  000000
12465
12466          073320                  DT53=DT52
12467
12468  073352  001632  001516  001636  DT54:   .WORD   $TMP0,$ERRPC,$TMP2,0
12469  073360  000000
12470
12471  073362  001632  001516  001636  DT136:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
12472  073370  001640  000000
12473
12474          073362                  DT137=DT136
12475
12476  073374  001632  001516  001636  DT140:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
12477  073402  001640  000000
12478
12479          073374                  DT141=DT140
12480
12481          073374                  DT142=DT140
12482
12483          073374                  DT143=DT140
12484
12485          073374                  DT144=DT140
12486
12487          073374                  DT145=DT140
12488
12489          073374                  DT146=DT140
12490
12491          073374                  DT147=DT140
12492
12493  073406  001632  001634  001636  DT150:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
12494  073414  001640  000000
12495
12496  073420  001632  001634  001636  DT151:  .WORD   $TMP0,$TMP1,$TMP2,0
12497  073426  000000
12498
12499          073420                  DT152=DT151
12500          073420                  DT153=DT151
12501  073430  001632  001634  001636  DT154:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
12502  073436  001640  000000
12503          073430                  DT155=DT154
12504          073430                  DT156=DT154
12505          073420                  DT157=DT151
12506          073420                  DT160=DT151
12507          073420                  DT161=DT151
12508
12509
12510  073442  000000  000000  000000  BOTTOM: .WORD   0,0,0
```

```
12511  101442                    BUTPRG=BOTTOM+6000
12512  000001                    .END
```

DEKBDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
AA     = 000011          3829#
AAADR1  014052           3870*  3871*  3880   3881   3991*  3992*  4013#
AAADR2  014062           3880*  3881*  3882*  3883*  3884*  3894   3896   3916   3917   3969   3970   3987   3988
                         4017#  4038   4040
AADONE  014230           3996   4066#
AAERGS  014050           3867*  3956   3981   4000*  4012#
AAERR1  014076           3952   4024#
AAERR2  014166           4045   4054#
AAEXER  014066           3868*  3997*  4019#  4028
AAFLG1  014042           3866*  3944   3968   3986   3995   3999*  4003#
AAFLG2  014044           3910*  3979*  4006#  4058   4061
AAGS    014046           3865*  3874   3998*  4010#
AAHIAD  013264           3854#
AALOAD  013262           3853#  3855#  3894   3896
AAOFST  014056           3859*  3860*  3882   3884   4015#
AATMP1  014070           4020#
AATMP2  014072           3857   4021#
AA1     013356           3870#  4001
AA16    013736           3985#
AA2     013376           3877#  3993
AA3     013470           3901   3906   3910#
AA4     013640           3941   3950#
AA5     013666           3967#
AA6     013712           3973#  4063
AA7     013762           3948   3991#  4060
AAB     014000           3904   3995#
ABORTT  043604           8903   9034#
ADRNG   053375           9080   10574#
AVMBL   037716           8128*  8134#  8154
BACKAD  043722           9050*  9059   9060#
BB     = 000010          3591#
BBADR1  012630           3621   3657   3659   3673   3694*  3695*  3711#  3732   3734   3759   3760   3763   3764
                         3766   3769   12416
BBADR2  012634           3713#  3763*  3764*  12419
BBADR3  012640           3715#  3765*  3766*  3767*  3768*  3769*  3770*  12419
BBCNT1  012650           3649*  3650*  3723#  3773*  3774*  12419
BBDONE  013140           3705   3788#
BBERR1  012660           3652   3729#  3785
BBERR2  012716           3739   3748#
BBERR3  012732           3749   3752#
BBERR4  013116           3776   3783#
BBERR5  013120           3782   3784#
BBFLG1  012644           3604*  3691   3701   3704   3706*  3718#  3753
BBFLG2  012646           3648*  3698   3720#  3772*
BBGM    012656           3606*  3613   3708*  3727#
BBGS    012654           3605*  3611   3617   3707*  3726#
BBHIAD  012144           3602#
BBLOAD  012142           3601#  3657   3659
BBO     012122           3592#  3609
BB1     012166           3608#  3709
BB2     012204           3611#  3615
BB3     012244           3621#
BB4     012412           3654#  3696
BB5     012450           3664   3669   3673#
BB6     012532           3689   3694#  3786
BB7     012546           3667   3698#
```

```
688     012574          3699    3704*
BIT0  = 000001          1300*   5827    5981
BIT00 = 000001          1290*   1300
BIT01 = 000002          1289*   1299
BIT02 = 000004          1288*   1298
BIT03 = 000010          1287*   1297
BIT04 = 000020          1286*   1296
BIT05 = 000040          1285*   1295
BIT06 = 000100          1284*   1294
BIT07 = 000200          1283*   1293
BIT08 = 000400          1282*   1292    8484
BIT09 = 001000          1281*   1291    8494    8553
BIT1  = 000002          1299*
BIT10 = 002000          1280*   8538
BIT11 = 004000          1279*   8501    9809    9968
BIT12 = 010000          1278*   9805    9814    9966    10429
BIT13 = 020000          1277*   8543
BIT14 = 040000          1276*   8470    9807    9816    9973    10431   10437
BIT15 = 100000          1275*
BIT2  = 000004          1298*
BIT3  = 000010          1297*
BIT4  = 000020          1296*
BIT5  = 000040          1295*   9785    9949    10152   10411
BIT6  = 000100          1294*   2220    9176    9736    9803    9893    10060   10158   10234   10366
BIT7  = 000200          1293*   9154    9602    9603    9610
BIT8  = 000400          1292*   9801    9964    10427
BIT9  = 001000          1291*
BOTPRG= 101442          2702    3091    3127    3132    3147    3216    3231    3255    3622    12511*
BOTTOM  073442          2211    9160    12510*  12511
BPTVEC= 000014          1307*
CACHVE= 000114          1315*   2289*   2414*   2514*   2644*   2727*   2835*   2927*   3032*   3112*   3184*   3315*   3608*
                        3652*   3776*   3785*   3834*   3927*   3952*   4030*   4099*   4150*   4327*   4337*   4393*   4446*
                        4623*   4633*   4675*   4808*   4947*   5106*   5278*   5284*   5342*   5526*   5642*   5649*   5769*
                        5777*   5821*   5831*   5874*   5975*   5985*   6028*   6118*   6204*   6374*   6713*   9012*   9019*
                        9048*
CALRH4= 104423          7694    7714    7734    7752    8239    8915*
CALRK5= 104424          7863    7883    7903    7921    8916*   9580
CALRP4= 104422          7530    7550    7570    7588    8285    8914*   9553
CALRS4= 104421          7362    7382    7402    7420    8262    8913*   9525
CALUBE= 104425          8009    8025    8917*
CC    = 000014          4670*
CCDONE  020064          4728    4772    4790*
CCERR1  017572          4675    4735*
CCERR2  017620          4739    4741*
CCERR3  020004          4688    4699    4711    4723    4775*
CCTMP1  017564          4730*
CCTMP2  017566          4679    4731*
CC1     017354          4682*
CC10    017516          4712    4716*   4771
CC11    017546          4723*   4726
CC12    017554          4722    4725*
CC13    017560          4724    4728*
CC2     017376          4688*   4691
CC3     017404          4687    4690*
CC4     017412          4689    4692*   4763
CC5     017440          4699*   4702
```

DEKBDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
CC6      017446            4698    4701#
CC7      017454            4700    4704#   4767
CC8      017504            4711#   4714
CC9      017512            4710    4713#
CHAINQ   044142            8439    9159#
CLEAN    043634            8902    9046#
CNRNG    053605            9094   10601#
CONCMS   052350            9158   10454#
CONFLG   044050            9090    9116#
CONFL2   044064            9122#
CONTRL=  177746            1324#   2320*   2544*   2756*   2950*   3114*   3186*   3390*   3392*   3611*   3613*   3619*   3874*
                           4116*   4118*   4124*   4216*   4326    4330*   4336*   4412*   4414*   4420*   4514*   4622    4626*
                           4632*   4677*   4810*   4971*   5131*   5369*   5374*   5380*   5447*   5460*   5536*   5538*   5543*
                           5560*   5653*   5655*   5665*   5678*   5830*   5876*   5984*   6030*   6145*   6147*   6155*   6168*
                           6376*   6411*   6414*   6417*   6483*   6485*   6487*   6489*   6491*   6493*   6495*   6497*   6502*
                           6504*   6506*   6508*   6510*   6512*   6514*   6516*   6520*   6522*   6539*   6541*   6559*   6561*
                           6578*   6580*   6604*   6685*   6689*   6739*   6742*   6745*   8328*   8330*   8336*   8343*   9055*
CPSPUR   043452            2221    2728    9004#   9049
CPUERR=  177766            1336#   8565*   9006    9057*   9425*   3436*   9454*   9464*   9474*
CR     = 000015            1212#   8674    8684
CRLF   = 000200            1213#   2196    8645    8684   10454   10456   10463   10473   10476   10483   10493   10503   10506
                          10516   10522   10525   10530   10537   10542   10549   10559   10562   10569   10574   10588   10601
                          10613   10627   10640   10650   10663   10668   10676   10682   10688   10693   10699   10705   10718
                          10730   10735   10741   10751   10757   10762   10768   10779   10812   10834   10848   10854   10864
                          10874   10886   10915   10923   10933   10987   11014   11036   11055   11076   11089   11102   11116
                          11143   11171   11193   11208   11232   11255   11279   11295   11320   11331   11347   11364   11373
                          11391   11398   11411   11433   11440   11453   11475   11482   11496   11518   11525   11538   11560
                          11567   11581   11602   11609   11622   11642   11649   11663   11684   11691   11705   11717
CYCNT    033726            7223    7252#
DD     = 000020            5273#
DDCNTR   022676            5288*   5304    5311    5318*   5325#
DDDONE   022700            5308    5313    5327#
DDISP  = 177570            1206#   1645    2176
DDPD     022546            5287    5294#
DDPER    022644            5284    5315#
DDPER1   022670            5316    5321#
DDPU1    022604            5303#
DDPU2    022630            5305    5310#
DDPV     022560            5295    5297#
DDTMP    022674            5286*   5298    5303    5323#
DD1      022512            5281    5284#
DF1      072011            1722   12137#
DF10   = 072023            1743   12155#
DF11     072033            1746   12157#
DF12     072044            1749   12161#
DF13     072052            1752   12164#
DF136    072264            2004   12266#  12269
DF137  = 072264            2007   12269#
DF14     072057            1755   12167#
DF140    072270            2010   12271#  12274   12276   12278   12280   12282   12284   12286
DF141  = 072270            2013   12274#
DF142  = 072270            2016   12276#
DF143  = 072270            2019   12278#
DF144  = 072270            2022   12280#
DF145  = 072270            2025   12282#
DF146  = 072270            2028   12284#
```

```
DF147 = 072270          2031   12286#
DF15    072064          1758   12170#
DF150   072274          2034   12288#
DF151   072300          2037   12291#  12293   12294
DF152 = 072300          2040   12293#
DF153 = 072300          2043   12294#
DF154   072303          2046   12296#  12299   12300   12301   12302   12303
DF155 = 072303          2049   12299#
DF156 = 072303          2052   12300#
DF157 = 072303         12301#
DF16    072066          1761   12172#
DF160 = 072303          2058   12302#
DF161 = 072303          2061   12303#
DF17    072074          1764   12175#  12184
DF2     072015          1725   12140#  12143   12145
DF20  = 072074          1767   12184#
DF21    072124          1770   12186#
DF22    072152          1773   12195#  12200   12210   12214
DF23    072155          1776   12197#  12202   12212   12216
DF24  = 072152          1779   12200#
DF25  = 072155          1782   12202#
DF26    072162          1785   12204#
DF27    072166          1788   12207#
DF3   = 072015          1728   12143#
DF30  = 072152          1791   12210#
DF31  = 072155          1794   12212#
DF32  = 072152          1797   12214#
DF33  = 072155          1800   12216#
DF34    072173          1803   12218#  12221
DF35  = 072173          1806   12221#
DF36    072200          1809   12223#
DF37    072204          1812   12226#
DF4   = 072015          1731   12145#
DF40  = 072216          1815   12239#
DF41    072212          1818   12230#  12233
DF42  = 072212          1821   12233#
DF43    072216          1824   12235#  12239
DF44    072226          1827   12241#  12247   12256   12258
DF45  = 072226          1830   12247#
DF46    072245          1833   12249#  12254   12260   12262
DF47  = 072245          1836   12254#
DF5     072023          1734   12147#  12151   12153   12155
DF50  = 072226          1839   12256#
DF51  = 072226          1842   12258#
DF52  = 072245          1845   12260#
DF53  = 072245          1848   12262#
DF54    072261          1851   12264#
DF6   = 072023          1737   12151#
DF7   = 072023          1740   12153#
DH1     067211          1722   11777#
DH10  = 067365          1743   11813#
DH11    067417          1746   11815#
DH12    067473          1749   11824#
DH13    067566          1752   11836#
DH136   071205          2004   12033#  12041
DH137 = 071205          2007   12041#
```

```
DH14     067641          1755   11845#
DH140    071252          2010   12043#  12050   12052   12054   12056   12058   12060   12062
DH141  = 071252          2013   12050#
DH142  = 071252          2016   12052#
DH143  = 071252          2019   12054#
DH144  = 071252          2022   12056#
DH145  = 071252          2025   12058#
DH146  = 071252          2028   12060#
DH147  = 071252          2031   12062#
DH15     067734          1758   11857#
DH150    071315          2034   12064#
DH151    071401          2037   12074#
DH152    071450          2040   12082#
DH153    071517          2043   12091#
DH154    071601          2046   12102#
DH155    071637          2049   12109#
DH156    071675          2052   12116#
DH16     067760          1761   11862#
DH160    071733          2058   12124#
DH161    071761          2061   12129#
DH17     070046          1764   11873#  11883
DH2      067264          1725   11786#  11799   11801
DH20   = 070046          1767   11883#
DH21     070126          1770   11885#
DH22     070201          1773   11894#  11913   11937   11941
DH23     070250          1776   11902#  11915   11939   11943
DH24   = 070201          1779   11913#
DH25   = 070250          1782   11915#
DH26     070340          1785   11917#
DH27     070405          1788   11925#
DH3    = 067264          1728   11799#
DH30   = 070201          1791   11937#
DH31   = 070250          1794   11939#
DH32   = 070201          1797   11941#
DH33   = 070250          1800   11943#
DH34     070477          1803   11945#  11955
DH35   = 070477          1806   11955#
DH36     070557          1809   11957#
DH37     070624          1812   11965#
DH4    = 067264          1731   11801#
DH40   = 070756          1815   11995#
DH41     070711          1818   11977#  11985
DH42   = 070711          1821   11985#
DH43     070756          1824   11997#  11995
DH44     071025          1827   11997#  12009   12018   12020
DH45   = 071025          1830   12009#
DH46     071120          1833   12011#  12015   12022   12024
DH47   = 071120          1836   12016#
DH5      067365          1734   11803#  11809   11811   11813
DH50   = 071025          1839   12018#
DH51   = 071025          1842   12020#
DH52   = 071120          1845   12022#
DH53   = 071120          1848   12024#
DH54     071144          1851   12026#
DH6    = 067365          1737   11809#
DH7    = 067365          1740   11811#
```

```
DISPLA  001542          1645#   2176*   2184*   8515*   8537*
DISPRE  000174          1601#   2184
DRH4T1  036256          7638*   7652    7655*   7665    7793#
DRH4T2  036260          7639*   7653    7656*   7666    7794#
DRK5T1  037070          7803*   7817    7820*   7830    7962#
DRK5T2  037072          7804*   7818    7821*   7831    7963#
DRP4T1  035464          7470*   7484    7487*   7497    7629#
DRP4T2  035466          7471*   7485    7488*   7498    7630#
DRRH4   035470          7269    7637#
DRRK5   036262          7270    7802#
DRRP4   034656          7268    7469#
DRRS4   034044          7267    7301#
DRS4T1  034652          7302*   7316    7319*   7329    7461#
DRS4T2  034654          7303*   7317    7320*   7330    7462#
DRUBE   037074          7271    7969#
DSWR  = 177570          1205#   1644    2175
DT1     072310          1722   12310#
DT10  = 072340          1743   12329#
DT11    072362          1746   12331#
DT12    072406          1749   12336#
DT13    072424          1752   12340#
DT136   073362          2004   12471#  12474
DT137 = 073362          2007   12474#
DT14    072440          1755   12343#
DT140   073374          2010   12476#  12479   12481   12483   12485   12487   12489   12491
DT141 = 073374          2013   12479#
DT142 = 073374          2016   12481#
DT143 = 073374          2019   12483#
DT144 = 073374          2022   12485#
DT145 = 073374          2025   12487#
DT146 = 073374          2028   12489#
DT147 = 073374          2031   12491#
DT15    072454          1758   12346#
DT150   073406          2034   12493#
DT151   073420          2037   12496#  12499   12500   12505   12506   12507
DT152 = 073420          2040   12499#
DT153 = 073420          2043   12500#
DT154   073430          2046   12501#  12503   12504
DT155 = 073430          2049   12503#
DT156 = 073430          2052   12504#
DT157 = 073420         12505#
DT16    072462          1761   12348#
DT160 = 073420          2058   12506#
DT161 = 073420          2061   12507#
DT17    072500          1764   12352#
DT2     072322          1725   12313#  12317   12319
DT20    072562          1767   12362#
DT21    072644          1770   12372#
DT22    072722          1773   12381#
DT23    072732          1776   12384#
DT24    072746          1779   12387#
DT25    072756          1782   12390#
DT26    072772          1785   12393#
DT27    073004          1788   12396#
DT3   = 072322          1728   12317#
DT30    073020          1791   12399#
```

```
DT31     073030            1794   12402#
DT32     073044            1797   12405#
DT33     073054            1800   12408#
DT34     073070            1803   12411#  12414
DT35  =  073070            1806   12414#
DT36     073104            1809   12416#
DT37     073116            1812   12419#
DT4   =  072322            1731   12319#
DT40  =  073144            1815   12432#
DT41     073132            1818   12423#  12426
DT42  =  073132            1821   12426#
DT43     073144            1824   12428#  12432
DT44     073166            1827   12434#  12441
DT45  =  073166            1830   12441#
DT46     073226            1833   12443#  12449
DT47  =  073226            1836   12449#
DT5      072340            1734   12321#  12325   12327   12329
DT50     073260            1839   12451#  12458
DT51  =  073260            1842   12458#
DT52     073320            1845   12460#  12466
DT53  =  073320            1848   12466#
DT54     073352            1851   12458#
DT6   =  072340            1737   12325#
DT7   =  072340            1740   12327#
DUBET1   037442            7969*   7988    7991*   7995    8059#
DUBET2   037444            7970*   7989    7992*   8060#
EE    =  000016            4941#
EEDONE   021550            5024    5068    5086#
EEERR1   021256            4947    5031#
EEERR2   021304            5035    5037#
EEERR3   021470            4983    4995    5007    5019    5071#
EETMP1   021250            5026#
EETMP2   021252            4973    5027#
EE1      021040            4977#
EE10     021202            5008    5012#   5067
EE11     021232            5019#   5022
EE12     021240            5018    5021#
EE13     021244            5020    5024#
EE2      021062            4983#   4986
EE3      021070            4982    4985#
EE4      021076            4984    4988#   5059
EE5      021124            4995#   4998
EE6      021132            4994    4997#
EE7      021140            4996    5000#   5063
EE8      021170            5007#   5010
EE9      021176            5006    5009#
EMTVEC=  000030            1310#   2159*   2160*
EM1      055371            1722   10793#
EM10     056331            1743   10886#
EM11     056422            1746   10897#
EM12     056510            1749   10907#
EM13     056635            1752   10923#
EM136    062605            2004   11331#
EM137    063022            2007   11356#
EM14     056715            1755   10933#
EM140    063240            2010   11382#
```

```
EM141    063601            2013   11424#
EM142    064141            2016   11466#
EM143    064503            2019   11509#
EM144    065044            2022   11551#
EM145    065276            2025   11593#
EM146    065727            2028   11633#
EM147    066262            2031   11675#
EM15     056754            1758   10940#
EM150    066614            2034   11717#
EM151    066700            2037   11727#  11737   11738
EM152 =  066700            2040   11737#
EM153 =  066700            2043   11738#
EM154    066761            2046   11740#
EM155    067013            2049   11746#
EM156    067045            2052   11752#
EM16     057024            1761   10948#
EM160    067112            2058   11761#
EM161    067144            2061   11767#
EM17     057100            1764   10957#  10967
EM2      055456            1725   10803#
EM20  =  057100            1767   10967#
EM21     057161            1770   10969#
EM22     057245            1773   10979#  11023   11065   11085
EM23     057461            1776   11005#  11025
EM24  =  057245            1779   11023#
EM25  =  057461            1782   11025#
EM26     057615            1785   11027#
EM27     057762            1788   11047#
EM3      055650            1728   10826#
EM30  =  057245            1791   11065#
EM31     060127            1794   11067#  11087
EM32  =  057245            1797   11085#
EM33  =  060127            1800   11087#
EM34     060266            1803   11089#
EM35     060372            1806   11102#
EM36     060501            1809   11116#
EM37     060633            1812   11133#
EM4      055764            1731   10840#
EM40     060715            1815   11143#
EM41     061070            1818   11163#
EM42     061254            1821   11185#
EM43     061444            1824   11208#
EM44     061572            1827   11224#  11303
EM45     061766            1830   11247#  11305
EM46     062165            1833   11271#  11307
EM47     062306            1836   11287#  11309
EM5      056077            1734   10854#
EM50  =  061572            1839   11303#
EM51  =  061766            1842   11305#
EM52  =  062165            1845   11307#
EM53  =  062306            1948   11309#
EM54     062432            1851   11311#
EM6      056157            1737   10864#
EM7      056237            1740   10874#
ERRNG    053505            9087   10588#
ERR/EC=  000004            1303#   2173    2174*   2185*   2728*   8475    8476*   8478*   8481*   9049*
```

```
ERTYPE  044334          3702    5718    8547    9037    9232#
ERT1    044446          9262#   9326
ERT2    044660          9266    9271    9281    9289    9295    9307    9316    9320#
ERT3    044664          9301    9323#
ERT4    044674          9260    9325    9328#
ERT5    044676          9240    9329#
FF    = 000015          4803#
FFDONE  020662          4863    4907    4925#
FFERR1  020370          4808    4870#
FFERR2  020416          4874    4876#
FFERR3  020602          4822    4834    4846    4858    4910#
FFTMP1  020362          4865#
FFTMP2  020364          4812    4866#
FF1     020140          4816#
FF10    020312          4847    4851#   4906
FF11    020342          4858#   4861
FF12    020350          4857    4860#
FF13    020356          4859    4863#
FF2     020164          4822#   4825
FF3     020172          4821    4824#
FF4     020202          4823    4827#   4898
FF5     020230          4834#   4837
FF6     020236          4833    4836#
FF7     020246          4835    4839#   4902
FF8     020276          4846#   4849
FF9     020304          4845    4848#
GETBUF  037726          7309    7477    7645    7810    7974    8142#
GETMP1  037724          8137#   8143#   8164
GGFLG1  024516          5529#   5553    5577    5584    5599    5608    5619#   5623#
GGGM    024522          5531#   5538    5558    5618#   5626#
GGGS    024520          5530#   5536    5541    5617#   5625#
GG1     024046          5529#
GG10    024430          5598    5604#
GG11    024464          5607    5614#
GG12    024524          5620    5628#
GG2     024070          5534#   5621
GG3     024100          5536#   5540
GG4     024156          5546    5547#   5557
GG5     024250          5564    5565#   5568
GG6     024276          5572    5573#   5591
GG7     024326          5576    5581#
GG8     024356          5582    5590#
GG9     024400          5594    5595#   5615
GNS   = ****** U        1600    2195    8892    8893    8894    8895    8896    8899    8900    8902    8903    8904    8905
                        8906    8907    8908    8909    8910    8911    8913    8914    8915    8916    8917
GTBIHI  037632          8120#   8122#   8123
GTBILO  037630          8119#   8121#   8124
GTBINT  037612          7189    8113#
GTMSIZ  037714          8126#   8133#
GTRNH   037722          8130#   8136#   8148    8152#
GTRNL   037720          8129#   8135#   8147    8150#
HH    = 000017          5098#
HHDONE  022450          5184    5228    5246#
HHERR1  022156          5106    5191#
HHERR2  022204          5195    5197#
HHERR3  022370          5143    5155    5167    5179    5231#
```

DEKBDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
HHTMP1  022150          5186#
HHTMP2  022152          5133    5187#
HH1     021726          5137#
HH10    022100          5168    5172#   5227
HH11    022130          5179#   5182
HH12    022136          5178    5181#
HH13    022144          5180    5184#
HH2     021752          5143#   5146
HH3     021760          5142    5145#
HH4     021770          5144    5148#   5219
HH5     022016          5155#   5158
HH6     022024          5154    5157#
HH7     022034          5156    5160#   5223
HH8     022064          5167#   5170
HH9     022072          5166    5169#
HIADRS= 177742          1322#   2451    2678    2873    3067    3758    4322    4618    4746    4881    5042    5202    5764
                        6325    9022
HIAFLG  044044          9076    9114#
HIAFL2  044060          9120#
HIMFLG  044054          9104    9118#
HIMFL2  044070          9124#
HITMIS= 177752          1326#   3464    3484    3688    3940    4200    4270    4496    4568    5385    5401    5417    5433
                        5551    5575    5597    5669    5693    5837    5883    5991    6037    6159    6423    6437    6452
                        6466    6526    6545    6565    6584    6751    6765    6780    6794    6827    6844    6863    6880
                        6899    6916    6935    6952    6977    7000    7023    7046    8210
HHRNG   054015          9108    10627#
HT    = 000011          1210#   8643    8664
IIA   = 000025          5816#
IIAR1   026044          5831    5924#
IIAR2   026104          5874    5934#
IIAT1   026042          5832    5870    5878    5916    5922#   5928    5938
IIA1    025540          5825#   5947
IIA2    025564          5828    5830#
IIA3    025666          5867#
IIA4    025706          5874#   5875    5933
IIA5    026020          5913#
IIA6    026040          5920#
IIA7    026140          5829    5920    5944#
IIA8    026156          5945    5949#
IIB   = 000026          5970#
IIBR1   026520          5985    6078#
IIBR2   026560          6028    6088#
IIBT1   026516          5986    6024    6032    6070    6076#   6082    6092
IIB1    026214          5979#   6101
IIB2    026240          5982    5984#
IIB3    026342          6021#
IIB4    026362          6028#   6029    6087
IIB5    026474          6067#
IIB6    026514          6074#
IIB7    026614          5983    6074    6098#
IIB8    026632          6099    6103#
INDONE  040044          7294    8171#
INTMP1  033724          7214#   7235    7249#
INTO    033516          7189#
INT1    033534          7196#
INT2    033560          7202#   7241
```

```
INT3     033606          7212#   7219
INT4     033624          7213    7217#
INT5     033640          7216    7223#
INT6     033706          7205    7221    7237#
IOTVEC=  000020          1308#   2157*   2158*
JJCNT    023766          5345*   5479*   5481    5487    5493    5499#
JJPAT1   023770          5356*   5366    5449    5478    5483*   5489*   5495*   5501#   12321
JJPAT2   023772          5354    5361*   5367    5455    5465    5466    5483    5484*    5502#  12321
JJPAT3   023774          5363*   5370    5371    5372    5391    5407    5489    5490*    5503#  12321
JJPAT4   023776          5364*   5375    5376    5377    5423    5439    5461    5495     5496*   5504#  12321
JJPAT5   024000          5464*   5467    5470*   5472    5478*   5484    5490    5496     5505#
JJTMP1   024002          5507#
JJTMP2   024004          5346    5508#
JJ1      023036          5365#   5476    5485    5491    5497
JJ10     023504          5445    5446#
JJ11     023540          5453    5454#
JJ12     023560          5456    5460#
JJ13     023614          5462    5470#
JJ14     023630          5468    5474#
JJ15     023706          5482    5487#
JJ16     023736          5488    5493#
JJ17     024014          5494    5512#
JJ2      023144          5382#
JJ3      023204          5386    5391#
JJ4      023234          5397    5398#
JJ5      023274          5402    5407#
JJ6      023324          5413    5414#
JJ7      023364          5418    5423#
JJ8      023414          5429    5430#
JJ9      023454          5434    5439#
KDPAR0=  172360          1478#
KDPAR1=  172362          1479#
KDPAR2=  172364          1480#
KDPAR3=  172366          1481#
KDPAR4=  172370          1482#
KDPAR5=  172372          1483#
KDPAR6=  172374          1484#
KDPAR7=  172376          1485#
KDPDR0=  172320          1456#
KDPDR1=  172322          1457#
KDPDR2=  172324          1458#
KDPDR3=  172326          1459#
KDPDR4=  172330          1460#
KDPDR5=  172332          1461#
KDPDR6=  172334          1462#
KDPDR7=  172336          1463#
KERSTK=  001100          1196#
KIPAR0=  172340          1467#   2295    2522    2732    2930    3199    3323    3630    3837    4127    4423    4949    5109
                         7171    9363
KIPAR1=  172342          1468#
KIPAR2=  172344          1469#
KIPAR3=  172346          1470#
KIPAR4=  172350          1471#
KIPAR5=  172352          1472#
KIPAR6=  172354          1473#   2402*   2632*   2822*   3019*   3222*   3237*   3261*   3458*   3478*   3680*   3925*   4193*
                         4265*   4490*   4563*   8094*
```

```
KIPAR7= 172356          1474#
KIPDR0= 172300          1445#   2297    2524    2734    2932    3201    3325    3632    3839    4129    4425    4951    5111
                        7173
KIPDR1= 172302          1446#
KIPDR2= 172304          1447#
KIPDR3= 172306          1448#
KIPDR4= 172310          1449#
KIPDR5= 172312          1450#
KIPDR6= 172314          1451#
KIPDR7= 172316          1452#
KKCNT1  031370          6407#   6625#   6627    6634    6641    6652#
KKERR1  031416          6536    6665#
KKERR2  031430          6556    6669#
KKERR3  031450          6575    6674#
KKERR4  031464          6595    6678#
KKERR5  031506          6667    6672    6676    6681    6683#
KKFLG1  031366          6382#   6477    6597    6599#   6603#   6649#
KKPAT1  031372          6377#   6484    6515    6521    6624    6630#   6637#   6644#   6655#   12377
KKPAT2  031374          6378#   6403    6488    6511    6540    6610    6611    6630    6631#   6656#   12377
KKPAT3  031376          6379#   6492    6507    6560    6637    6638#   6657#   12377
KKPAT4  031400          6380#   6496    6503    6579    6606    6644    6645#   6658#   12377
KKPAT5  031402          6609#   6612    6615#   6617    6624#   6631    6638    6645    6659#
KKTMP1  031404          6661#
KKTMP2  031406          6388    6662#
KK1     030024          6376#
KK10    031132          6594    6597#
KK11    031152          6598    6603#
KK12    031214          6607    6615#
KK13    031230          6613    6619#
KK14    031240          6620    6623#
KK15    031306          6628    6634#
KK16    031336          6635    6641#
KK17    031534          6642    6692#
KK2     030064          6388#
KK3     030170          6409    6410#   6600    6621    6632    6639    6646
KK4     030516          6480    6500#
KK5     030626          6499    6519#
KK6     030704          6535    6538#
KK7     030770          6555    6558#
KK8     031046          6574    6577#
KSP    =%000006         1231#
LF     = 000012         1211#   8678    8684
LKS    = 177546         1207#
LKVEC  = 000100         1314#
LLCNT1  025224          5686#   5706#   5717    5734#
LLERR1  025232          5740#   5777
LLERR2  025436          5700    5780#
LLERR3  025416          5769    5776#
LLFLG1  025212          5645#   5671    5695    5714    5722#   5726#   5742    5760    5786
LLFLG2  025214          5687#   5707#   5711    5727#
LLFLG4  025216          5689#   5704    5729#   5753#   5780#
LLGM    025222          5647#   5653    5676    5720#   5732#
LLGS    025220          5646#   5655    5663    5721#   5731#
LLMER   025226          5736#   5740#   5745    5766
LLTMP1  025230          5738#   5741#   5744#   5745
LL1     024556          5645#
```

```
LL10    025504          5723    5794#
LL2     024600          5648#   5650    5724
LL3     024624          5653#   5658
LL4     024774          5682#
LL5     025024          5688    5689#   5709
LL6     025062          5694    5698#
LL7     025072          5699    5702#   5778    5793
LL8     025112          5705    5708#
LL9     025164          5712    5720#
LOADRS= 177740          1321#   2427    2429    2450    2655    2657    2677    2849    2851    2872    3043    3045    3066
                        3732    3734    3757    3779    4038    4040    4321    4328    4617    4624    4738    4745    4873
                        4880    5034    5041    5194    5201    5749    5763    5928    5938    6082    6092    6324    9016
                        9021
LOADFLG 044042          9074    9113#
LOADFL2 044056          9119#
LOC   = 026464          5854#   5855#   5856#   5857    5903#   5904#   5905#   5906    6008#   6009#   6010#   6011    6057#
                        6058#   6059#   6060
LOOP    004330          2205#   8449
MAINT = 177750          1325#   2412    2642    2833    3030    3955    5848    5894    6002    6048    9052*
MANFLG  044052          9097    9117#
MANFL2  044066          9123#
MAPH0 = 170202          1561#
MAPH00= 170202          1497#   1561    2816*   2818*   3013*   3015*   3221*   3236*   3260*
MAPH01= 170206          1499#   1563
MAPH02= 170212          1501#   1565
MAPH03= 170216          1503#   1567
MAPH04= 170222          1505#   1569
MAPH05= 170226          1507#   1571
MAPH06= 170232          1509#   1573
MAPH07= 170236          1511#   1575
MAPH1 = 170206          1563#
MAPH10= 170242          1513#
MAPH11= 170246          1515#
MAPH12= 170252          1517#
MAPH13= 170256          1519#
MAPH14= 170262          1521#
MAPH15= 170266          1523#
MAPH16= 170272          1525#
MAPH17= 170276          1527#
MAPH2 = 170212          1565#
MAPH20= 170302          1529#
MAPH21= 170306          1531#
MAPH22= 170312          1533#
MAPH23= 170316          1535#
MAPH24= 170320          1537#
MAPH25= 170326          1539#
MAPH26= 170332          1541#
MAPH27= 170336          1543#
MAPH3 = 170216          1567#
MAPH30= 170342          1545#
MAPH31= 170346          1547#
MAPH32= 170352          1549#
MAPH33= 170356          1551#
MAPH34= 170362          1553#
MAPH35= 170366          1555#
MAPH36= 170372          1557#
```

```
MAPH37= 170376           1559#
MAPH4 = 170222           1569#
MAPH5 = 170226           1571#
MAPH6 = 170232           1573#
MAPH7 = 170236           1575#
MAPL0 = 170200           1560#
MAPL00= 170200           1496#   1560    2815*   2817*   3012*   3014*   3220*   3235*   3259*   4954    5114    10268
MAPL01= 170204           1498#   1562
MAPL02= 170210           1500#   1564
MAPL03= 170214           1502#   1566
MAPL04= 170220           1504#   1568
MAPL05= 170224           1506#   1570
MAPL06= 170230           1508#   1572
MAPL07= 170234           1510#   1574
MAPL1 = 170204           1562#
MAPL10= 170240           1512#
MAPL11= 170244           1514#
MAPL12= 170250           1516#
MAPL13= 170254           1518#
MAPL14= 170260           1520#
MAPL15= 170264           1522#
MAPL16= 170270           1524#
MAPL17= 170274           1526#
MAPL2 = 170210           1564#
MAPL20= 170300           1528#   10127
MAPL21= 170304           1530#
MAPL22= 170310           1532#
MAPL23= 170314           1534#
MAPL24= 170320           1536#
MAPL25= 170324           1538#
MAPL26= 170330           1540#
MAPL27= 170334           1542#
MAPL3 = 170214           1566#
MAPL30= 170340           1544#
MAPL31= 170344           1546#
MAPL32= 170350           1548#
MAPL33= 170354           1550#
MAPL34= 170360           1552#
MAPL35= 170364           1554#
MAPL36= 170370           1556#
MAPL37= 170374           1558#
MAPL4 = 170220           1568#
MAPL5 = 170224           1570#
MAPL6 = 170230           1572#
MAPL7 = 170234           1574#
MEMERR= 177744           1323#   2445*   2448    2455*   2673*   2675    2682*   2868*   2871    2877*   3062*   3065    3071*
                         3748    3756    3784*   4028    4054*   4310    4318    4334*   4606    4614    4630*   4735    4747
                         4758*   4870    4882    4893*   5031    5043    5054*   5191    5203    5214*   5315    5317*   5740
                         5776*   5925    5932*   5935    5942*   6079    6086*   6089    6096*   6323    8564*   9013    9020
                         9056*
MMDES   044236           8904    9192#
MMERR1  033260           6836    6853    6872    6889    7073#
MMERR2  033272           6908    6925    6944    6961    7077#
MMERR3  033304           7075    7080#
MMERR4  033322           6990    7013    7086#
MMERR5  033342           7036    7059    7091#
```

```
MMERR6  033362          7089    7095*
MMESRS  052354          9171   10456*
MMPAT1  033236          6717*   6819    6988    7063*  12358   12368
MMPAT2  033240          6718*   7011    7064*  12358
MMPAT3  033242          6719*   7034    7065*  12358   12368
MMPAT4  033244          6720*   7057    7066*  12358   12368
MMRFLG  044046          9083    9115*
MMRFL2  044062          9121*
MMR0   = 177572         1346*   1350    2310*   2405*   2457*   2537*   2635*   2689*   2748*   2826*   2878*   3023*   3078*
                        3215*   3398*   3507*   3646*   3863*   4143*   4439*   4969*   5129*   7186*   9053*   9352
MMR1   = 177574         1347*   1351
MMR2   = 177576         1348*   1352
MMR3   = 172516         1349*   1353    2309*   2404*   2458*   2536*   2634*   2690*   2747*   2824*   2879*   3021*   3079*
                        3214*   3397*   3508*   3645*   3862*   4142*   4438*   4968*   5128*   7187*   9054*
MMSKIP= 104411          2294    2516    2730    2928    3185    3317    3598    3835    4102    4398    4946    5104    8904*
MMTMP1  033246          7068*
MMTMP2  033250          6722    7069*
MMVEC  = 000250         1317*
MM1     031570          6717*
MM10    032242          6835    6837*
MM11    032300          6845    6850*
MM12    032316          6852    6854*
MM13    032360          6864    6869*
MM14    032376          6871    6873*
MM15    032434          6881    6886*
MM16    032452          6888    6890*
MM17    032514          6900    6905*
MM18    032532          6907    6909*
MM19    032570          6917    6922*
MM2     031706          6736    6738*
MM20    032606          6924    6926*
MM21    032650          6936    6941*
MM22    032666          6943    6945*
MM23    032724          6953    6958*
MM24    032742          6960    6962*
MM25    033024          6978    6988*
MM26    033036          6989    6993*
MM27    033076          7001    7011*
MM28    033110          7012    7016*
MM29    033150          7024    7034*
MM3     031772          6752    6760*
MM30    033162          7035    7039*
MM31    033222          7047    7057*
MM32    033400          7058    7061    7101*
MM4     032030          6766    6775*
MM5     032056          6781    6789*
MM6     032122          6795    6804    6805*
MM7     032132          6807*   6809
MM8     032144          6812*   6814
MM9     032224          6828    6833*
MNRNG   053707          9101   10613*
MONF    044234          2205*   9164*   9180*
MONTTY  044232          2208*   9172    9178*
MSIZER  044260          8905    9208*
MTA101  054276         10663*
MTA11   052544         10483*  12331
```

DEKBDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
MTA120  054403          10676#
MTA124  054664          10718#
MTA126  054756          10730#
MTA131  055036          10741#
MTA134  055201          10762#
MTA135  055235          10768#
MTA17   052611          10491#  10514   12355   12365
MTA20   052645          10500#  12362
MTA21   052654          10503#  12372
MTA43   052741          10516#  12428
MTA45   053014          10525#  12434   12443
MTA5    052462          10473#  12321
MTA50   053072          10537#  12451   12460
MTA77   054120          10640#
MTB120  054433          10682#
MTB126  055004          10735#
MTB131  055120          10751#
MTB135  055265          10774#
MTB17   052616          10493#  12352   12362
MTB21 = 052611          10514#  12374
MTB45   053042          10530#  12434   12437   12443   12446   12451   12454   12460   12463
MTB77   054134          10643#
MTC120  054463          10688#
MTC131  055153          10757#
MTC135  055307          10779#
MTC17   052636          10497#  12352
MTC45   053057          10534#  12437   12443   12446   12454   12460   12463
MTC77   054176          10650#
MTD120  054513          10693#
MTD135  055343          10785#
MTD77   054233          10656#
MTE120  054543          10699#
MTF120  054573          10705#
MTG120  054623          10711#
MOM1  = 000014           1589#  2320    2544    2756    2950    4677    4810    4971    5131    6376    6483    6487    6491
                         6495   6502    6506    6510    6514    6520    6539    6559    6578    6604    6685
M1M0  = 000014           1588#  1589    3114    3186    5447
NNDEV   040724           8196   8349#
NNDONE  041172           8237   8393    8408#
NNDO    040742           8355#
NND2    041022           8353   8373#
NND3    041106           8375   8391#
NNGRM   040602           8199*  8234*   8311#   8328    8341
NNGRPF  040604           8198*  8213    8230*   8231    8312#
NNGRS   040600           8200*  8233*   8310#   8330    8334
NNRH4   040320           8239#  8395
NNRH4U  040323           8241#  8401*
NNRP4   040504           8220   8285#   8377
NNRP4U  040507           8287#  8383*
NNRS4   040412           8216   8262#   8357
NNRS4U  040415           8264#  8364*
NNSTUP  040614           8202   8323#   8324
NNUD    040576           8203   8216    8220    8308#   8351*   8357*   8377*   8395*
NN1     040114           8196#  8260    8283    8306
NN2     040120           8198#  8365    8384    8402
NN3     040140           8202#  8235
```

```
NN5     040260          8219    8223    8225    8230#
NN6     040314          8232    8237#
NOCNC   044214          9156    9175#
PARCNT  044072          5826    5980    9134#
PC    =%000007          1234#   2416*   2646*   2837*   3034*   3368*   3376*   3384*   3574*   3702*   3964*   3983*   4688*
                        4699*   4711*   4723*   4787*   4822*   4834*   4846*   4858*   4922*   4993*   4995*   5007*   5019*
                        5083*   5143*   5155*   5167*   5179*   5243*   5718*   5826*   5980*   6536*   6556*   6575*   6595*
                        6690*   6836*   6853*   6872*   6889*   6908*   6925*   6944*   6961*   6990*   7013*   7036*   7059*
                        7189*   7191*   7309*   7318*   7373*   7379*   7393*   7398*   7413*   7417*   7430*   7435*   7459*
                        7477*   7486*   7541*   7547*   7561*   7566*   7581*   7585*   7598*   7603*   7627*   7645*   7654*
                        7705*   7711*   7725*   7730*   7745*   7749*   7762*   7767*   7791*   7810*   7819*   7874*   7880*
                        7894*   7899*   7914*   7918*   7931*   7936*   7960*   7974*   7990*   8018*   8022*   8034*   8038*
                        8057*   8098*   8131*   8149*   8168*   8191*   8202*   8203*   8251*   8274*   8297*   8345*   8426*
                        8429*   8439*   8443*   8448    8547*   8652*   8659*   8666*   8680*   8682*   8864*   8954    8988*
                        9037*   9141*   9167*   9276*   9294*   9306*   9312*   9330*   9375*   9383*   9419*   9421*   9430*
                        9432*   9440*   9447*   9450*   9458*   9460*   9468*   9470*   9478*   9511*   9543*   9571*   9598*
                        9606*   9611*   9718*   9719*   9720*   9722*   9740*   9746*   9761*   9763*   9769*   9772*   9779*
                        9783*   9815*   9823*   9830*   9876*   9877*   9879*   9880*   9898*   9904*   9919*   9921*   9927*
                        9930*   9944*   9948*   9972*   9980*   9987*   10036*  10037*  10039*  10042*  10065*  10070*  10084*
                        10086*  10093*  10099*  10120*  10136*  10160*  10166*  10174*  10220*  10221*  10239*  10245*  10256*
                        10258*  10263*  10281*  10289*  10295*  10302*  10349*  10350*  10351*  10371*  10377*  10393*  10395*
                        10402*  10405*  10409*  10435*  10444*  10451#
PDMSG1  053120          5291    10542#
PDMSG2  053276          5307    10562#
PIRQ  = 177772          1204#
PIRQVE= 000240          1316#
POWERM  052413          8953    10463#
PP    = 000005          3107*   3171
PPHIAD  010106          3117#   3121
PPLIM   010274          3120*   3123    3125*   3129    3144    3160    3164#
PPLOAD  010104          3116#   3123    3125
PP1     010074          3114#
PP2     010142          3122    3124    3127#
PP3     010162          3133#   3145
PP4     010212          3135    3143#
PP5     010226          3148#   3161
PP6     010262          3152    3159#
PP7     010276          3162    3166#
PR0   = 000000          1237#
PR1   = 000040          1238#
PR2   = 000100          1239#
PR3   = 000140          1240#
PR4   = 000200          1241#
PR5   = 000240          1242#
PR6   = 000300          1243#
PR7   = 000340          1244#
PS    = 177776          1201#   1202
PSW   = 177776          1202#   9058*   9701*   9859*   10018*  10204*  10333*
FWRVEC= 000024          1309#   2163*   2164*   8922*   8923*   8932*   8938*   8950*   8951*
QQERR1  027674          6204    6323#
QQERR2  027752          6331    6334#
QQERR3  027764          6335    6338#
QQERR4  027766          6333    6337    6339#
QQFLG1  027360          6139*   6161    6232#   6312*   6329
QQFLG2  027356          6125*   6230#   6317*
QQGM    027374          6132*   6145    6166    6238#   6310*
```

```
$DGS    027372          6131*   6147    6153    6237#   6311*
$DHI    027354          6177*   6186*   6195*   6198*   6219    5224    6227#
$DLO    027352          6176*   6185*   6194*   6197*   6223    6226#
$DPAT1  027362          6121*   6156    6233#   6316*   6334
$DTMP1  027364          6219*   6220*   6221    6234#
$DTMP2  027366          6235#
$DTMP3  027370          6236#
$$1     026672          6129#   6319
$$10    027376          6170    6186    6188    6194    6198    6246#
$$11    027452          6172    6259#
$$12    027454          6176    6179    6190    6195    6197    6266#
$$13    027530          6181    6279#
$$14    027532          6177    6195    6286#
$$15    027606          6299#
$$16    027610          6255    6275    6295    6301#   6339
$$17    027612          6304#
$$18    027654          6313    6316#
$$19    027772          6318    6341#
$$2     026722          6143#   6314
$$3     027006          6156#   6165
$$4     027050          6160    6165#
$$5     027130          6175    6179#
$$6     027172          6184    6188#
$$7     027234          6193    6197#
$$8     027250          6178    6187    6196    6200#
$$9     027300          6208    6212#   6308
$$9.5   027312          6217#
RANDTP  037610          8078*   8081    8110#
RANDWR  037446          8077#   8911
RESMON  044114          2217    9151#
RESREG= 104406          7215    7220    7351    7519    7683    7852    7999    8105    8166    8344    8900#   8987    9380
                        9477    9510    9738    9742    9824    9895    9900    9981    10062   10067   10167   10236   10241
                        10296   10368   10373   10445
RESVEC= 000010          1304#
RHWT    037604          8087*   8091    8101*   8108#
RHHAA   035514          7642#   7774
RHHAA1  035734          7672*   7687#
RHHAA2  035736          7679*   7688#
RHHAA3  035732          7640*   7686#   7690*
RHHAE   003676          2118#   9445*   9446*   10356*
RHHAS   003662          2111#
RHHASS  033760          7269#
RHHBA   003650          2106#   10355*
RHHBB   035755          7661*   7696#
RHHCC   035756          7667*   7697#
RHHCLR  052332          10393   10444   10448#
RHHCR   033734          7230    7256#   7291    7769*   7781*
RHHCS1  003644          2104#   9439    9445    10365*  10384   10415   10418*  10420   10449
RHHCS2  003654          2108#   9442*   10353*  10383*  10389   10413*  10414*  10417*  10440   10448*
RHHCS3  003700          2119#
RHHCYL  051520          10326#
RHHDA1  051500          10318#  10340*  10357
RHHDA2  051502          10319#  10341*  10407*
RHHDB   003666          2113#
RHHDD   035760          7698#
RHHDFL  045442          8194    8258*   9411*   9482#   9610*
```

```
R44DR   003664          2112#  10357*
R44DT   003672          2115#   9443
R44EE   035762          7673*   7699#
R44ER   003660          2110#  10391   10442
R44ER1  051460          7777    8249    9405*  10310#  10346*  10392*  10443*
R44ER2  051462          7782    8253   10311#  10347*  10389*  10440*
R44ER3  051464          7784    8254   10312#  10348*  10390*  10441*
R44ER4  051466          7783    8255   10313#  10391*  10442*
R44FF   035764          7680*   7700#
R44FLG  051456          9400*  10309#  10328   10364*  10372*  10376*
R44FT   040610          8194*   8259*   8318#   8392    8400*   8407*
R44FUN  051474         10316#  10338*  10363
R44GG   036006          7702    7710#
R44HAN  051522          8915   10328#
R44HH   036021          7662*   7716#
R44HH1  051534         10329   10333#
R44HH2  051650         10353#
R44HH3  051760         10367   10371#
R44HH4  051774         10361   10376#
R44HH5  052010         10371   10377   10382#
R44HH51 052012         10383#  10398   10400
R44HH6  052076         10386   10388   10397#
R44II   036022          7669*   7717#
R44JJ   036024          7718#
R44KK   036026          7674*   7719#
R44LL   036030          7681*   7720#
R44MA1  051504         10320#  10342*  10355
R44MA2  051506         10321#  10343*  10356
R44MM   036052          7722    7729#
R44MR1  003670          2114#  10358*
R44MR2  003652          2107#
R44NN   036065          7663*   7736#
R44OO   036066          7670*   7737#
R44PP   036070          7738#
R44QQ   036072          7675*   7739#
R44RB   033772          7275#   7646    7647
R44RDY  052140         10350   10411#
R44REG  003642          2103#   9441
R44REX  003674          2117#   9448
R44RR   036074          7677*   7740#
R44SEC  051516         10325#
R44SS   036160          7760    7766#
R44ST   003656          2109#  10390   10399   10426   10441
R44SUN  033746          7262#   7659
R44S1   052114         10349   10404#
R44S2   052124         10351   10407#
R44TMP  051472         10315#  10334*  10337
R44TRK  051514         10324#
R44UNI  051476         10317#  10339*  10353   10383   10404*  10412   10414   10417
R44USE  051470         10314#  10411*  10412*  10413   10448
R44V    003744          2142#  10360
R44VEC  051512         10323#  10345*  10378
R44WC   003646          2105#  10354*
R44WCT  051510         10322#  10344*  10354   10408*
R44XX   036176          7770    7773#
R44YY   036204          7705    7711    7725    7730    7745    7749    7762    7767    7776#
```

DEKBDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
RHHZZ   036254          7772    7791#
RHH111  036116          7742    7749#
RHH112  036127          7660*   7754#
RHH113  036130          7668*   7755#
RHH114  036132          7756#
RHH115  036134          7676*   7757#
RHH116  036136          7678*   7758#
RKSAA   036306          7807#   7943
RKSAA1  036546          7841*   7856#
RKSAA2  036550          7848*   7857#
RKSAA3  036544          7805*   7855#   7859*
RKSASS  033762          7270#
RKSBA   003714          2126#   10049*
RKSBB   036567          7826#   7865#
RKSCC   036570          7832*   7866#
RKSCLR  050702          10084   10166   10170#
RKSCR   033736          7257#   7283    7938*   7950*
RKSCS1  003710          2124#   10046*  10050*  10057*  10078   10139*  10140   10144*  10146   10171*  10172
RKSCYL  047752          10011#  10109*
RKSDA   003716          2127#   10047*  10051*  10138*  10143*  10170*
RKSDA1  047732          10003#  10025*  10051   10101   10119*
RKSDA2  047734          10004#  10026*
RKSDB   003720          2128#
RKSDD   036572          7836*   7867#
RKSDFL  045443          9412*   9483#   9597*
RKSDS   003704          2122#   9457    10082   10090   10151   10164
RKSEE   036574          7842*   7868#
RKSER   003706          2123#   10081   10154   10163
RKSER1  047712          7946    9406*   9590    9995#   10032*  10083*  10165*
RKSER2  047714          7951    9996#   10033*  10081*  10163*
RKSER3  047716          7953    9997#   10034*  10082*  10164*
RKSER4  047720          7952    9998#
RKSFF   036576          7849*   7869#
RKSFLG  047710          9401*   9994#   10013   10056*  10066*  10071*
RKSFT   040611          8319#
RKSFUN  047726          10001#  10023*  10055
RKSGG   036620          7871    7879#
RKSHAN  047754          8916    10013#
RKSHH   036633          7827*   7885#
RKSH1   047766          10014   10018#
RKSH2   050106          10046#
RKSH3   050212          10061   10065#
RKSH4   050226          10053   10070#
RKSH5   050254          10065   10070   10077#
RKSH51  050256          10078#  10089   10091
RKSH6   050320          10080   10088#
RKSII   036634          7834*   7886#
RKSJJ   036636          7837*   7887#
RKSKK   036640          7843*   7888#
RKSLL   036642          7850*   7889#
RKSMA1  047736          10005#  10027*  10049   10124   10135*
RKSMA2  047740          10006#  10028*  10050   10125   10134*
RKSMM   036664          7891    7898#
RKSNN   036677          7828*   7905#
RKSOO   036700          7835*   7906#
RKSPP   036702          7839*   7907#
```

```
RK5QQ   036704          7844*  7908#
RK5RB   033774          7276#  7811    7812
RK5RDY  050542         10037   10138#
RK5REG  003702          2121#  9459
RK5RR   036706          7846*  7909#
RK5SEC  047750         10010#  10110*
RK5SS   036772          7929   7935#
RK5SUN  033750          7263#  7824
RK5S1   050336         10036   10095#
RK5S2   050360         10039   10101#
RK5S3   050462         10042   10122#
RK5TMP  047724         10000#  10019*  10022
RK5TRK  047746         10009#  10111*
RK5UNI  047730         10002#  10024*  10047   10095   10098*  10138   10143
RK5USE  047722          9999#
RK5V    003746          2143#  10052   10072*
RK5VEC  047744         10008#  10030*  10074
RK5WC   003712          2125#  10048*
RK5WCT  047742         10007#  10029*  10048   10122*
RK5XX   037010          7939   7942#
RK5YY   037016          7874   7880    7894    7899    7914    7918    7931    7936    7945#
RK5ZZ   037066          7947   7960#
RK5111  036730          7911   7918#
RK5112  036741          7825*  7923#
RK5113  036742          7833*  7924#
RK5114  036744          7838*  7925#
RK5115  036746          7845*  7926#
RK5116  036750          7847*  7927#
RLWT    037606          8086*  8092    8095    8100*   8109#
RP4AA   034702          7474#  7610
RP4AA1  035142          7508*  7523#
RP4AA2  035144          7515*  7524#
RP4AA3  035140          7472*  7522#  7526*
RP4AS   003604          2087#
RP4ASS  033756          7268#
RP4BA   003572          2082#  9726*
RP4BAE  003636          2100#  9727*
RP4BB   035163          7493*  7532#
RP4CC   035164          7499*  7533#
RP4CCC  003624          2095#
RP4CLR  046736          9761   9823    9827#
RP4CR   033732          7227   7255#  7287    7605*   7617*
RP4CS1  003566          2080#  9429    9735*   9752    9789    9792*   9793    9828
RP4CS2  003576          2084#  9724*   9751*   9757    9787*   9788*   9791*   9819    9827*
RP4CS3  003640          2101#
RP4CYL  046054          9695#
RP4DA   003574          2083#  9728*
RP4DA1  046034          9687#  9708*   9729    9774    9777*
RP4DA2  046036          9688#  9709*   9728    9778*
RP4DB   003610          2089#
RP4DC   003622          2094#  9729*
RP4DD   035166          7503*  7534#
RP4DFL  045441          8193   8304*   9410*   9481#   9570*
RP4DS   003600          2085#  9758    9766    9799    9820
RP4DT   003614          2091#
RP4EC1  003632          2098#
```

```
RP4EC2  DC3634          2099#
RP4EE   035170          7509*   7535#
RP4ER1  046014          7613    8295    9404*   9563    9679#   9714*   9760*   9822*
RP4ER2  046016          7618    8299    9680#   9715*   9757*   9819*
RP4ER3  046020          7620    8300    9681#   9716*   9758*   9820*
RP4ER4  046022          7619    8301    9682#   9759*   9821*
RP4FF   035172          7516*   7536#
RP4FLG  046012          9399*   9678#   9697    9734*   9741*   9745*
RP4FT   040607          8193*   8305*   8317#   8374    8389*
RP4FUN  046030          9685#   9706*   9733
RP4GG   035214          7538    7546#
RP4HAN  046056          8914    9697#
RP4HH   035227          7494*   7552#
RP4HI   046070          9698    9701#
RP4H2   046210          9724#
RP4H3   046316          9737    9740#
RP4HN   046332          9731    9745#
RP4H5   046346          9740    9746    9750#
RP4H51  046350          9751#   9765    9767
RP4H6   046434          9754    9756    9764#
RP4II   035230          7501*   7553#
RP4JJ   035232          7504*   7554#
RP4KK   035234          7510*   7555#
RP4LA   003606          2088#
RP4LL   035236          7517*   7556#
RP4MA1  046040          9689#   9710*   9726
RP4MA2  046042          9690#   9711*   9727    9782*
RP4MM   035260          7558    7565#
RP4MR   003612          2090#
RP4NN   035273          7495*   7572#
RP4OF   003620          2093#   9814*
RP4OO   035274          7502*   7573#
RP4PP   035276          7506*   7574#
RP4QQ   035300          7511*   7575#
RP4RB   033770          7274#   7478    7479
RP4RDY  046522          9719    9785#
RP4REG  003564          2079#   9431
RP4RR   035302          7513*   7576#
RP4RR1  003602          2086#   9759    9821
RP4RR2  003626          2096#
RP4RR3  003630          2097#
RP4SEC  046052          9694#
RP4SN   003616          2092#
RP4SS   035366          7596    7602#
RP4SUN  033744          7261#   7491
RP4S1   046452          9718    9771#
RP4S2   046462          9720    9774#
RP4S3   046506          9722    9781#
RP4TMP  046026          9684#   9702*   9705
RP4TRK  046050          9693#
RP4UNI  046032          9686#   9707*   9724    9751    9771*   9786    9788    9791
RP4USE  046024          9683#   9785*   9786*   9787    9827
RP4V    003742          2141#   9730
RP4VEC  046046          9692#   9713*   9747
RP4WC   003570          2081#   9725*
RP4WCT  046044          9691#   9712*   9725    9781*
```

```
RP4XX   035404          7606    7609#
RP4YY   035412          7541    7547    7561    7566    7581    7585    7598    7603    7612#
RP4ZZ   035462          7614    7627#
RP4111  035324          7578    7585#
RP4112  035335          7492*   7590#
RP4113  035336          7500*   7591#
RP4114  035340          7505*   7592#
RP4115  035342          7512*   7593#
RP4116  035344          7514*   7594#
RR    = 000003          2722#
RRADR1  007136          2760*   2761*   2766    2768    2784    2785    2891*   2892*   2898#
RRADR2  007142          2784*   2785*   2786*   2787*   2788*   2796    2798    2815    2816    2849    2951    2900#  12399
                       12402
RRADR3  007146          2758*   2759*   2766    2768    2786    2788    2880    2882    2884*   2887*   2888*   2893*   2894*
                        2902#
RRDONE  007152          2803    2806    2905#
RRHIAD  006406          2752#
RRLOAD  006404          2751#   2754*   2796    2798
RR1     006446          2763#   2896
RR10    007100          2881    2883    2887#
RR11    007112          2773    2778    2891#
RR12    007132          2885    2889    2896#
RR2     006504          2776    2782#
RR3     006540          2793#
RR4     006576          2808    2811#
RR5     006722          2840#
RR6     006730          2835    2846#
RR7     006766          2856    2865#
RR8     007006          2859    2861    2871#
RR9     007044          2843    2869    2878#
RSET  = 104407          2483    2705    2905    3094    3166    3284    3576    3788    4066    4359    4653    4790    4925
                        5086    5246    5327    5949    6103    6692    7168    8171    8189    8408    8902#   9038    9152
                        9196
RS4AA   034070          7306#   7442
RS4AA1  034330          7340*   7355#
RS4AA2  034332          7347*   7356#
RS4AA3  034326          7304*   7354#   7358*
RS4AS   003546          2071#
RS4ASS  033754          7199    7266    7267#
RS4BA   003534          2066#   9884*
RS4BAE  003560          2076#   9885*
RS4BB   034351          7325*   7364#
RS4CC   034352          7331*   7365#
RS4CLR  047672          9919    9980    9984#
RS4CR   033730          7197    7224    7254#   7289    7437*   7449*
RS4CS1  003530          2064#   9418    9892*   9910    9953    9956*   9957    9984*   9985
RS4CS2  003540          2068#   9882*   9909*   9915    9951*   9952*   9955*   9976
RS4CS3  003562          2077#
RS4CYL  047016          9853#   9938*
RS4DA   003536          2067#   9886*
RS4DA1  046776          9845#   9866*   9886    9932    9943*
RS4DA2  047000          9846#   9867*
RS4DB   003552          2073#
RS4DD   034354          7335*   7366#
RS4DFL  045440          7196    8192    8281*   9409*   9480#   9542*
RS4DS   003542          2069#   9916    9924    9963    9977
```

RS4DT	003556	2075#								
RS4EE	034356	7341*	7367#							
RS4ER	003544	2070#	9917	9978						
RS4ER1	046756	7445	8272	9403*	9535	9837#	9872*	9918*	9979*	
RS4ER2	046760	7450	8276	9838#	9873*	9915*	9976*			
RS4ER3	046762	7452	8277	9839#	9874*	9916*	9977*			
RS4ER4	046764	7451	8278	9840#	9917*	9978*				
RS4FF	034360	7348*	7368#							
RS4FLG	046754	9398*	9836#	9855	3891*	9899*	9903*			
RS4FT	040606	8192*	8282*	8316#	8352	8363*	8371*	8362*		
RS4FUN	046772	9843#	9864*	9890						
RS4GG	034402	7370	7378#							
RS4HAN	047020	8913	9855#							
RS4HH	034415	7326*	7384#							
RS4H1	047032	9856	9859#							
RS4H2	047152	9882#								
RS4H3	047252	9894	9898#							
RS4H4	047266	9888	9903#							
RS4H5	047302	9898	9904	9908#						
RS4H51	047304	9909#	9923	9925						
RS4H6	047370	9912	9914	9922#						
RS4II	034416	7333*	7385#							
RS4JJ	034420	7336*	7386#							
RS4KK	034422	7342*	7387#							
RS4LA	003550	2072#								
RS4LL	034424	7349*	7388#							
RS4MA1	047002	9847#	9868*	9884						
RS4MA2	047004	9848#	9869*	9885	9947*					
RS4MM	034446	7390	7397#							
RS4MR	003554	2074#								
RS4NN	034461	7327*	7404#							
RS4OO	034462	7334*	7405#							
RS4PP	034464	7338*	7406#							
RS4QQ	034466	7343*	7407#							
RS4RB	033766	7273#	7310	7311	8113	8156				
RS4RDY	047500	9877	9949#							
RS4REG	003526	2063#	9420							
RS4RR	034470	7345*	7408#							
RS4SEC	047014	9852#								
RS4SS	034354	7428	7434#							
RS4SUN	033742	7198	7260#	7323						
RS4S1	047406	9876	9929#							
RS4S2	047416	9879	9932#							
RS4S3	047464	9880	9946#							
RS4TMP	046770	9842#	9860*	9863						
RS4TRK	047012	9851#	9937*							
RS4UNI	046774	9844#	9865*	9882	9909	9929*	9950	9952	9955	
RS4USE	046766	9841#	9949*	9950*	9951	9984				
RS4V	003740	2140#	9887							
RS4VEC	047010	9850#	9871*	9905						
RS4WC	003532	2065#	9883*							
RS4WCT	047006	9849#	9870*	9883	9946*					
RS4XX	034572	7438	7441#							
RS4YY	034600	7373	7379	7393	7398	7413	7417	7430	7435	7444#
RS4ZZ	034650	7446	7459#							
RS4111	034512	7410	7417#							

```
RS4112  034523          7324*   7422*
RS4113  034524          7332*   7423*
RS4114  034526          7337*   7424*
RS4115  034530          7344*   7425*
RS4116  034532          7346*   7426*
R0     =%000000         1216#   1224    2210*   2214*   2295*   2301*   2302*   2303*   2304*   2305*   2306*   2307*   2308*
                        2398*   2399*   2400*   2407*   2408*   2409*   2416    2522*   2528*   2529*   2530*   2531*   2532*
                        2533*   2534*   2535*   2628*   2629*   2630*   2637*   2638*   2639*   2646    2732*   2738*   2739*
                        2740*   2741*   2742*   2743*   2744*   2745*   2820*   2828*   2829*   2830*   2837    2930*   2936*
                        2937*   2938*   2939*   2940*   2941*   2942*   2943*   3017*   3025*   3026*   3027*   3034    3127*
                        3128*   3129    3132*   3133    3134    3136    3139    3143*   3144    3147*   3148    3149    3155
                        3159*   3160    3199*   3205*   3206*   3207*   3208*   3209*   3210*   3211*   3212*   3216*   3220
                        3226    3227*   3228    3231*   3235    3242    3244    3247    3251*   3252    3255*   3259    3266
                        3272    3277*   3278    3323*   3329*   3330*   3331*   3332*   3333*   3334*   3335*   3336*   3550*
                        3563*   3568*   3569*   3570    3609*   3612    3614    3617*   3618*   3619    3621*   3622*   3623*
                        3624*   3625*   3626*   3627*   3630*   3636*   3637*   3638*   3639*   3640*   3641*   3642*   3643*
                        3673*   3675    3677    3678    3837*   3843*   3844*   3845*   3846*   3847*   3848*   3849*   3850*
                        3857*   3858*   3859    3920*   3921*   3922*   3934*   3935*   3936*   3938    3939    3945    3964
                        3980    3983    4113*   4117    4119    4127*   4133*   4134*   4135*   4136*   4137*   4138*   4139*
                        4140*   4214*   4215*   4216    4409*   4413    4415    4418*   4419*   4420    4423*   4429*   4430*
                        4431*   4432*   4433*   4434*   4435*   4436*   4512*   4513*   4514    4679*   4680*   4681    4682*
                        4683*   4686    4690    4693*   4694*   4697    4701    4705*   4706*   4709    4713    4717*   4718*
                        4721    4725    4748    4750    4753    4754    4760    4777    4779    4782    4783    4812*   4813*
                        4814    4816*   4817*   4820    4824    4828*   4829*   4832    4836    4840*   4841*   4844    4848
                        4852*   4853*   4856    4860    4883    4885    4888    4889    4895    4912    4914    4917    4918
                        4949*   4961*   4962*   4965*   4973*   4974*   4975    4977*   4978*   4981    4985    4989*   4990*
                        4993    4997    5001*   5002*   5005    5009    5013*   5014*   5017    5021    5044    5046    5049
                        5050    5056    5073    5075    5078    5079    5109*   5121*   5122*   5125*   5133*   5134*   5135
                        5137*   5138*   5141    5145    5149*   5150*   5153    5157    5161*   5162*   5165    5169    5173*
                        5174*   5177    5181    5204    5206    5209    5210    5216    5233    5235    5238    5239    5299*
                        5301    5346*   5347*   5349    5351    5353    5384*   5391    5394    5400*   5407    5410    5416*
                        5423    5426    5432*   5439    5442    5448*   5449    5451    5454*   5455    5457    5534*   5537
                        5539    5541*   5542*   5543    5545*   5557*   5558*   5559*   5560    5563*   5568*   5571*   5591*
                        5593*   5615*   5650*   5654    5656    5657*   5661*   5675*   5676*   5677*   5678    5681*   5709*
                        5823*   5850    5864    5869    5896    5910    5915    5944    5946*   5977*   6004    6018    6023
                        6050    6064    6069    6098    6100*   6217*   6218*   6221    6388*   6389*   6391    6393    6406*
                        6619*   6623*   6722*   6723*   6724    6726    6815    6817    6966    6968    7171*   7177*   7178*
                        7179*   7180*   7181*   7182*   7183*   7184*   7196*   7204    7207    7209*   7219*   7237*   7312*
                        7313*   7345    7346    7347    7348    7349    7480*   7481*   7513    7514    7515    7516    7517
                        7648*   7649*   7677    7678    7679    7680    7681    7813*   7814*   7846    7847    7848    7849
                        7850    7978*   7979*   7983    7984    7985    8081*   8082    8083    8084    8085    8113*   8116*
                        8123*   8125*   8153*   8154*   8206*   8209    8214    8227*   8324*   8329    8334*   8335*   8336
                        8341*   8342*   8343    8352*   8361    8374*   8380    8392*   8398    8436*   8440*   8443    8586
                        8611*   8635    8636*   8637    8640*   8776    8786*   8790    8806    8807    8820*   8843    8846*
                        8849*   8853*   8856*   8859    8863*   8876    8877*   8878    8879*   8880*   8881*   8882*   8924
                        8949*   8977*   8981    8991*   8993*   8995*   9013*   9014    9136    9151*   9154*   9155    9208
                        9210*   9211*   9212*   9215*   9216*   9218*   9221    9222    9225*   9234    9235*   9236*   9242*
                        9243*   9244*   9245    9251    9258    9261*   9262    9267    9273    9283    9290    9296    9303
                        9309    9323*   9329*   9354*   9355*   9356*   9363*   9365    9504*   9505    9506    9507*   9520*
                        9540*   9548*   9568*   9575*   9595*   9705*   9706    9707    9708    9709    9710    9711    9712
                        9713    9730*   9731*   9732*   9733*   9734    9735    9736    9750    9752*   9753    9755    9762*
                        9764    9768*   9775*   9776*   9799*   9801    9803    9805    9807    9809    9811    9863*   9864
                        9865    9866    9867    9868    9869    9870    9871    9887*   9888*   9889*   9890*   9891    9892
                        9893    9908    9910*   9911    9913    9920*   9922    9926*   9933*   9934*   9935*   9936*   9937
                        9939*   9940*   9941*   9942    9963*   9964    9966    9968    9970    10022*  10023   10024   10025
                       10026   10027   10028   10029   10030   10052*  10053*  10054*  10055*  10056   10057   10060   10077
```

```
                    10078*  10079   10085*  10088   10092*  10095*  10096*  10097*  10098   10102*  10103*  10104*  10105*
                    10106   10107*  10108*  10111   10114*  10115*  10117*  10118*  10119   10124*  10129   10131*  10208*
                    10210   10211   10212   10213   10214   10215   10216   10228*  10229*  10230*  10231*  10232   10233
                    10234   10248   10249*  10250   10257*  10260   10262*  10265*  10271   10273*  10285*  10287   10337*
                    10338   10339   10340   10341   10342   10343   10344   10345   10360*  10361*  10362*  10363*  10364
                    10365   10366   10382   10384*  10385   10387   10394*  10397   10401*  10426*  10427   10429   10431
                    10433
R1      =%000001     1217#   1225    2211*   2213*   2296*   2299    2411*   2523*   2526    2641*   2733*   2736    2832*
                     2931*   2934    3029*   3133*   3134    3138    3148*   3151    3154    3200*   3203    3223*   3226*
                     3238*   3241    3250*   3262*   3265    3276*   3324*   3327    3389*   3391    3393    3400*   3409
                     3411    3429    3453    3455    3456    3467    3468    3487    3488    3498*   3499    3546*   3548*
                     3549*   3551    3552    3558*   3559*   3560*   3561*   3562*   3567*   3570*   3571    3610*   3615*
                     3631*   3634    3838*   3841    4114*   4120*   4122*   4123*   4124    4128*   4131    4149*   4209*
                     4222*   4281*   4410*   4416*   4424*   4427    4445*   4507*   4520*   4579*   4681*   4685*   4692*
                     4696*   4704*   4708*   4716*   4720*   4738    4755    4761    4765    4769    4784    4814*   4819*
                     4827*   4831*   4839*   4843*   4851*   4855*   4873    4890    4896    4900    4904    4919    4957*
                     4958*   4959    4975*   4980*   4988*   4992*   5000*   5004*   5012*   5016*   5034    5051    5057
                     5061    5065    5080    5117*   5118*   5119    5135*   5140*   5148*   5152*   5160*   5164*   5172*
                     5176*   5194    5211    5217    5221    5225    5240    5300*   5302*   5348*   5349*   5370*   5371*
                     5372*   5373    5384    5388    5400    5404    5535*   5540*   5544*   5548*   5549    5550    5554
                     5556    5562*   5566*   5567    5570*   5574    5578    5581    5585    5587    5590*   5592*   5596
                     5600    5604    5611    5614    5651*   5658*   5660*   5666*   5667    5668    5672    5680*   5691*
                     5692    5696    5708*   5749    5762    5773*   5774    5787    5849*   5865    5895*   5911    6003*
                     6019    6049*   6065    6201*   6217    6249*   6250    6251*   6252    6253*   6254    6255*   6256
                     6257    6269*   6270    6271*   6272    6273*   6274    6275*   6276    6277    6289*   6290    6291*
                     6292    6293*   6294    6295*   6296    6297    6304*   6328    6391*   6392*   6396    6397    6412
                     6413    6420    6426    6434    6440    6486*   6490*   6513*   6517*   6523    6529    6542    6548
                     6665    6669    6724*   6725*   6729    6730    6740    6741    6748    6754    6762    6768    6807*
                     6815*   6816*   6826    6829    6837*   6843    6846    6854*   6862    6865    6873*   6879    6882
                     6890*   6966*   6967*   6972    6974    6980    6995    6997    7003    7074    7087    7172*   7175
                     7197*   7202*   7210*   7214    7217*   7223*   7224    7226*   7227    7229*   7230    7232*   7238
                     7311*   7340    7341    7342    7343    7344    7479*   7508    7509    7510    7511    7512    7647*
                     7672    7673    7674    7675    7676    7812*   7841    7842    7843    7844    7845    7977*   7980
                     7981    7982    8082*   8088    8114*   8117*   8124*   8126    8127*   8128    8151*   8152    8158*
                     8160    8165    8207*   8228*   8325*   8332*   8337*   8339    8356*   8361    8363    8367*   8376*
                     8380    8382    8386*   8394*   8398    8400    8404*   8587    8610*   8777    8790*   8791    8795
                     8819*   8844    8847*   8850*   8854*   8855*   8857*   8858*   8860    8862*   8925    8948*   8973*
                     8977    8978*   8990*   8992*   8994*   9016*   9017*   9018    9134*   9136    9139*   9160*   9162
                     9209    9213*   9217*   9224*   9257    9258*   9264    9269    9275    9285    9292    9298    9311
                     9324    9328*   9347*   9348    9360*   9361*   9362*   9369    9505*   9509*   9521*   9537    9538*
                     9549*   9565    9566*   9576*   9592    9593*   9774*   9777    9793*   9794    9796    9816    9932*
                     9938    9942*   9943    9957*   9958    9960    9973   10101*  10110   10116*  10117   10125*  10126*
                    10130   10132*  10146*  10148   10151*  10152   10156   10158   10266*  10267*  10272   10274*  10420*
                    10421   10423   10437
R10     =%000000     1224#
R11     =%000001     1225#
R12     =%000002     1226#
R13     =%000003     1227#
R14     =%000004     1228#
R15     =%000005     1229#
R2      =%000002     1218#   1226    2212*   2213    2297*   2299*   2394*   2396*   2401*   2412*   2524*   2526*   2624*
                     2626*   2631*   2642*   2734*   2736*   2833*   2932*   2934*   3030*   3149*   3150*   3151    3153
                     3201*   3203*   3241*   3242    3246    3265*   3268    3271    3325*   3327*   3423*   3429    3437
                     3439    3473    3475    3476    3489    3490    3493*   3495    3566*   3572*   3632*   3634*   3839*
                     3841*   3917*   3919*   3924*   3955*   4129*   4131*   4186*   4188    4190    4191    4258*   4260
                     4262    4263    4425*   4427*   4483*   4485    4487    4488    4556*   4558    4560    4561    4760*
```

```
                    4761    4764*   4765    4766*   4769    4895*   4896    4899*   4900    4903*   4904    4951*   4956*
                    4966*   5056*   5057    5060*   5061    5064*   5065    5111*   5116*   5126*   5216*   5217    5220*
                    5221    5224*   5225    5350*   5351*   5375*   5376*   5377*   5378    5416    5420    5432    5436
                    5574*   5581    5586    5596*   5606    5610    5663*   5664*   5665    5682*   5691    5698    5754
                    5755*   5756    5757*   5761    5782    5783*   5784    5785*   5788    5827    5848*   5863*   5865*
                    5894*   5909*   5911*   5981    6002*   6017*   6019*   6048*   6063*   6065*   6200*   6214*   6301*
                    6393*   6394*   6399    6400    6415    6416    6449    6455    6463    6469    6494*   6498*   6505*
                    6509*   6562    6568    6581    6587    6674    6678    6726*   6727*   6732    6733    6743    6744
                    6777    6783    6791    6797    6812*   6817*   6818*   6898    6901    6909*   6915    6918    6926*
                    6934    6937    6945*   6951    6954    6962*   6968*   6969*   7018    7020    7026    7041    7043
                    7049    7078    7092    7173*   7175*   7198*   7203*   7211*   7212    7218*   7235*   7239    7323*
                    7324    7325    7326    7327    7491*   7492    7493    7494    7495    7659*   7660    7661    7662
                    7663    7824*   7825    7826    7827    7828    8083*   8086    8091*   8093*   8095*   8096*   8097*
                    8099*   8156*   8160    8326*   8331    8338*   8340*   8358*   8364    8366*   8378*   8383    8385*
                    8396*   8401    8403*   8588    8609*   8778    8789*   8793*   8796    8803*   8804*   8805    8810*
                    8818*   8845    8848*   8851*   8861*   8926    8947*   8979*   8980    8981*   8996*   8997*   9135*
                    9138*   9161*   9162*   9366*   9367*   9370*   9372    9506*   9507    9508*   9522*   9537*   9542
                    9550*   9565*   9570    9577*   9592*   9597    10106*  10109   10112*  10113*  10118   10127*  10129*
                   10130*  10268*  10271*  10272*
R3      =%000003    1219#   1227    2298*   2300*   2393*   2395*   2398    2402    2525*   2527*   2623*   2625*   2628
                    2632    2735*   2737*   2933*   2935*   3202*   3204*   3266*   3267*   3268    3270    3326*   3328*
                    3453*   3454*   3459*   3462    3463    3473*   3474*   3479*   3483    3633*   3635*   3675*   3676*
                    3681*   3685    3686    3840*   3842*   3916*   3918*   3920    3925    4130*   4132*   4188*   4189*
                    4194*   4197    4198    4260*   4261*   4266*   4269    4426*   4428*   4485*   4486*   4491*   4494
                    4495    4558*   4559*   4564*   4567    4952*   4964*   5112*   5124*   5352*   5353*   5366*   5367*
                    5448    5454    5604*   5605*   5606    5609    5683*   5716    5754*   5782*   6143*   6146    6148
                    6149*   6153*   6154*   6155    6166*   6167*   6168    6212*   6484*   6486    6488*   6490    6492*
                    6494    6496*   6498    6503*   6505    6507*   6509    6511*   6513    6515*   6517    6521*   6534
                    6540*   6554    6560*   6573    6579*   6593    6683    6805*   6807    6808*   6812    6813*   6819*
                    6838*   6855*   6858*   6874*   6891*   6894*   6910*   6927*   6930*   6946*   6963*   7174*   7176*
                    7199*   7234    7239    7329*   7331    7332    7333    7334    7497*   7499    7500    7501    7502
                    7665*   7667    7668    7669    7670    7830*   7832    7833    7834    7835    7995*   7996    7997
                    8084*   8087    8092*   8094    8157*   8162*   8359*   8368*   8379*   8387*   8397*   8405*   8438*
                    8589    8608*   8718    8727*   8733*   8734*   8737*   8742*   8743*   8744    8753*   8779    8787*
                    8788*   8802*   8805*   8814*   8815*   8817*   8927    8946*   8976*   8989*   8996    9153*   9165
                    9365*   9369*   9373    10128*  10133*  10269*  10275*
R4      =%000004    1220#   1228    2549*   2552*   2560    2691    2954*   2957*   2965    3080    3456*   3457*   3476*
                    3477*   3678*   3679*   3956*   3957*   3981*   3982*   4191*   4192*   4263*   4264*   4488*   4489*
                    4561*   4562*   4953*   4957    4961    4963*   5113*   5117    5121    5123*   5354*   5461    5464
                    5465*   5466*   5467*   5470    5471*   5472*   5684*   5715    5756*   5784*   5847*   5863    5893*
                    5909    6001*   6017    6047*   6063    6144*   6150*   6151*   6165*   6170*   6171*   6174    6179*
                    6180*   6183    6188*   6189*   6192    6213*   6523*   6534    6542*   6554    6562*   6573    6581*
                    6593    6806*   6809*   6811*   6814*   7200*   7241*   7330*   7335    7336    7337    7338    7498*
                    7503    7504    7505    7506    7666*   7831*   7836    7837    7838    7839    8085*   8102*   8164*
                    8165*   8590    8607*   8719    8721*   8722*   8723*   8724    8725*   8739    8741*   8749*   8752*
                    8928    8945*   8975*   8982*   9159*   9163*
R5      =%000005    1221#   1229    2560*   2686*   2965*   3075*   3455*   3458    3475*   3478    3677*   3680    3779*
                    3780*   3781    4190*   4193    4262*   4265    4328*   4329*   4331    4487*   4490    4560*   4563
                    4624*   4625*   4627    4954*   4959*   4960*   5114*   5119*   5120*   5344*   5474*   5477*   5685*
                    5692*   5698    5789    5832*   5833    5834    5840    5850*   5864    5878*   5879    5880    5886
                    5896*   5910    5986*   5987    5988    5994    6004*   6018    6032*   6033    6034    6040    6050*
                    6064    6152*   6156*   6157    6158    6162    6172*   6173*   6174    6181*   6182*   6183    6190*
                    6191*   6192    6202*   6306*   6403*   6606    6609    6610*   6611*   6612*   6615    6616*   6617*
                    6822*   6825*   6826*   6833*   6834    6842*   6843*   6850*   6851    6861*   6862*   6869*   6870
                    6878*   6879*   6886*   6887    6897*   6898*   6905*   6906    6914*   6915*   6922*   6923    6933*
                    6934*   6941*   6942    6950*   6951*   6958*   6959    6972*   6988    6995*   7011    7018*   7034
```

```
                    7041*  7057   7081   7096   7207*  7212   8591   8606*  8720   8726*  8728*  8730*  8731*
                    8732*  8733   8751*  8780   8782*  8784*  8791*  8795*  8810   8816*  8929   8944*  8974*
                    8980*  8985*  8986
R6     =%000006     1222#  1230   2151*  2152*  2153
R7     =%000007     1223#  1234
SAVREG= 104405       7208   7308   7476   7644   7809   7973   8080   8146   8323   8699#  8972   9346   9408
                     9503   9704   9862   10021  10207  10336
SDPAR0= 172260       1434#
SDPAR1= 172262       1435#
SDPAR2= 172264       1436#
SDPAR3= 172266       1437#
SDPAR4= 172270       1438#
SDPAR5= 172272       1439#
SDPAR6= 172274       1440#
SDPAR7= 172276       1441#
SDPDR0= 172220       1412#
SDPDR1= 172222       1413#
SDPDR2= 172224       1414#
SDPDR3= 172226       1415#
SDPDR4= 172230       1416#
SDPDR5= 172232       1417#
SDPDR6= 172234       1418#
SDPDR7= 172236       1419#
SETBLE= 033754       7266#
SETMP   045516       9501*  9504   9513#
SETREG  045460       9419   9430   9440   9447   9458   9468   9501#
SIPAR0= 172240       1423#
SIPAR1= 172242       1424#
SIPAR2= 172244       1425#
SIPAR3= 172246       1426#
SIPAR4= 172250       1427#
SIPAR5= 172252       1428#
SIPAR6= 172254       1429#
SIPAR7= 172256       1430#
SIPDR0= 172200       1401#
SIPDR1= 172202       1402#
SIPDR2= 172204       1403#
SIPDR3= 172206       1404#
SIPDR4= 172210       1405#
SIPDR5= 172212       1406#
SIPDR6= 172214       1407#
SIPDR7= 172216       1408#
SIZDEV  045046       7191   8191   9398#
SIZE  = 104412       2311   2538   2750   2945   3115   3187   3339   3600   3852   4104   4400   8118   8905#
SIZEHI= 177762       1333#
SIZELO= 177760       1331#  9211
SIZRH4  046002       9450   9610#
SIZRK5  045670       9460   9575#
SIZRP4  045604       9432   9548#
SIZRS4  045520       9421   9520#
SIZTM1  045446       9415*  9476   9487#
SIZTM2  045450       9488#
SIZTM3  045452       9489#
SIZTM4  045454       9490#
SIZTM5  045456       9491#
SIZUBE  045754       9470   9602#
```

DEKBDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SKAD    043632          2286* 2511* 2724* 2924* 3109* 3181* 3312* 3594* 3831* 4097* 4391* 4672* 4805*
                        4943* 5100* 5275* 5282  5340* 5524* 5640* 5818* 5972* 6115* 6369* 6711* 7166*
                        8187* 9039  9041* 9177  9197
SKBADR  043724          8906  9074*
SKBCNR  043766          8908  9090*
SKBERR  043750          8907  9083*
SKBHMR  044022          8910  9104*
SKBMNR  044004          8909  9097*
SKIPT = 104410          5844  5890  5998  6044  8903* 9009  9027  9111
SKPBAD= 104413          8906*
SKPBCN= 104415          8908*
SKPBER= 104414          8907*
SKPBHM= 104417          8910*
SKPBMN= 104416          8909*
SKRNG   044036          9081  9088  9095  9102  9110*
SP    =%000006          1230* 1231  1232  1233  2155* 2173* 2181* 2185  2443  2444  2452  2453  2671
                        2672  2679  2680  2865  2866  2867  2874  2875  3059  3060  3061  3068  3069
                        3754  3755  3783  4024* 4025* 4056  4315  4316  4333  4611  4612  4629  4741
                        4743  4775  4876  4878  4910  5037  5039  5071  5197  5199  5231  5297* 5328*
                        5758  5759  5931  5941  6085  6095  6326  6327  6684  7080  7095  7193* 7194*
                        7233* 7234* 7374* 7394* 7414* 7431* 7455  7542* 7562* 7582* 7599* 7623  7706*
                        7726* 7746* 7763* 7787  7875* 7895* 7915* 7932* 7956  8019* 8035* 8054  8078
                        8079* 8143  8144* 8256  8279  8302  8433* 8475* 8478  8480  8481  8486* 8487*
                        8488  8511  8512  8516* 8542  8555* 8558* 8586* 8587* 8588* 8589* 8590* 8591*
                        8592* 8593* 8594* 8595* 8602* 8603* 8604* 8605* 8606  8607  8608  8609  8610
                        8611  8635* 8636  8637* 8639  8640  8641* 8643  8645  8647  8653  8655* 8657*
                        8665* 8669  8673  8674  8678  8710* 8711  8712  8713* 8718* 8719* 8720* 8726
                        8751  8752  8753  8754* 8755* 8776* 8777* 8778* 8779* 8780* 8781* 8782  8785*
                        8798  8800* 8802  8812  8814  8816  8817  8818  8819  8820  8822* 8823* 8843*
                        8844* 8845* 8861  8862  8863  8876* 8877  8924* 8925* 8926* 8927* 8928* 8929*
                        8930* 8931  8939* 8943  8944  8945  8946  8947  8948  8949  8954* 8973  8986*
                        9004  9007  9023  9024  9028  9034  9036  9050  9051* 9110  9195  9208* 9209*
                        9210  9222* 9224  9225  9234* 9238* 9257* 9264* 9269* 9275* 9277* 9278  9285*
                        9292* 9311* 9313  9328  9329  9347  9374* 9376* 9377  9381* 9424  9435  9453
                        9463  9473  9501  9502* 9702  9703* 9750* 9762  9768  9818  9860  9861* 9908*
                        9920  9926  9975  10019 10020* 10077* 10085 10092 10162 10205 10206* 10248* 10257
                        10262 10291 10334 10335* 10382* 10394 10401 10439
SPUR    043500          2222  2289  2514  2727  2927  3112  3184  3315  3608  3742  3744  3750  3834
                        3927  4030  4031  4048  4050  4099  4312  4393  4608  4737  4740  4872  4875
                        5033  5036  5193  5196  5278  5321  5342  5526  5642  5649  5747  5751  5821
                        5927  5937  5975  6081  6091  6118  6374  6713  9012* 9019  9048
SR0   = 177572          1350*
SR1   = 177574          1351*
SR2   = 177576          1352*
SR3   = 172516          1353*
SS    = 000004          2922*
SSADR1  010020          2952* 2953* 2958* 2959* 2960* 2968  2969  3084*
SSADR2  010024          2968* 2969* 2970* 2971* 2974  2976  2992  2994  3012  3013  3043  3045  3086*
                        12405 12408
SSCNST  010034          2974  2976  3091*
SSHIAD  007276          2947*
SSLOAD  007274          2946* 2948* 2992  2994
SSMASK  010030          2962* 2963* 2970  2971  3073* 3074* 3088*
SSP   =%000006          1232*
SS1     007314          2952*
SS10    007756          2986  3002  3038  3063  3073*
```

```
SS11    010040          3081    3094*
SS2     007334          2957*   3082
SS3     007354          2955    2962*
SS4     007376          2966    2968*   3077
SS5     007464          2981    2984    2989*
SS6     007522          2999    3004    3008*
SS7     007644          3040*
SS8     007702          3032    3050    3059*
SS9     007720          3053    3055    3065*
STACK = 001500          1195*   1196    1197    1198    1593*   2155    5297    5328    9051
START   003752          1604    2148*   8955
STKLMT= 177774          1203*
STOP    044200          9166    9170*
SUPSTK= 000700          1197*
SWR     001540          1644*   2153    2175*   2177    2183*   5280    8470    8484    8486    8494    8501    8538    8545
                        8550    8553    8930    8943*   9192
SWREG   000176          1602*   2183
SW0   = 000001          1272*
SW00  = 000001          1262*   1272
SW01  = 000002          1261*   1271
SW02  = 000004          1260*   1270
SW03  = 000010          1259*   1269
SW04  = 000020          1258*   1268
SW05  = 000040          1257*   1267
SW06  = 000100          1256*   1266
SW07  = 000200          1255*   1265
SW08  = 000400          1254*   1264
SW09  = 001000          1253*   1263
SW1   = 000002          1271*
SW10  = 002000          1252*
SW11  = 004000          1251*
SW12  = 010000          1250*   5280
SW13  = 020000          1249*
SW14  = 040000          1248*
SW15  = 100000          1247*
SW2   = 000004          1270*
SW3   = 000010          1269*
SW4   = 000020          1268*
SW5   = 000040          1267*
SW6   = 000100          1266*
SW7   = 000200          1265*   9192
SW8   = 000400          1264*
SW9   = 001000          1263*
SYSTID= 177764          1335*
SOMOM1= 000034          1567*   3605    3708    4109    4289    4405    4585    5369    5531    5617    6411    6739
SOM1  = 000030          1585*   3349    3513    3865    5647    5721    5830    5984    6132    6311    8200    8234
SIMO  = 000044          1584*   3350    3512    3998    5646    5720    5876    6030    6131    6310    8199    8233
SIMOM1= 000054          1586*   3606    3707    4110    4288    4406    4586    5374    5530    5618    6414    6742
TAB   = 000011          1583*   10471   10476   10483   10497   10500   10506   10516   10525   10534   10537   10663   10668
                        10730   10735   10741   10751   10757   10762   11777   11786   11794   11803   11815   11824   11836
                        11845   11851   11857   11862   11867   11873   11879   11885   11894   11902   11908   11917   11925
                        11945   11957   11965   11970   11977   11987   11997   12003   12011   12026   12033   12043   12064
                        12077   12085   12096   12102   12109   12116   12124   12129
TBITVE= 000014          1305*
TESTR1= 140000          1590*   5348    5544    5562    5570    5592    5660    5680    6201    6392    6725    6816    6967
                        8206    8244    8267    8290    8337
```

```
TESTR2= 142000          1591#   5350    6152    6200    6212    6394    6727    6815    6969    8326
TESTR3= 144000          1592#   5352
TKVEC = 000060          1312#   2208    2217*   2218*   9172*
TOP     004364          2206    2215#
TPVEC = 000364          1313#
TRAPVE= 000034          1311#   2161*   2162*
TRTVEC= 000014          1306#
TST1    004434          2282#
TST10   012112          3312    3589#
TST11   013142          3594    3827#
TST12   014232          3831    4092#
TST13   015654          4097    4386#
TST14   017302          4391    4668#
TST15   020066          4672    4801#
TST16   020664          4805    4939#
TST17   021552          494?    5096#
TST2    005342          2286    2507#
TST20   022452          5100    5272#
TST21   022706          5275    5337#
TST22   024014          5340    5521#
TST23   024524          5524    5637#
TST24   025504          5640    5814#
TST25   026160          5818    5968#
TST26   026634          5972    6113#
TST27   027772          6115    6366#
TST3    006242          2511    2720#
TST30   031536          6369    6708#
TST31   033400          6711    7164#
TST32   040046          7166    8185#
TST4    007154          2724    2920#
TST5    010042          2924    3105#
TST6    010300          3109    3177#
TST7    010740          3181    3309#
TT    = 000006          3179#
TTHIAD  010346          3189#   3192
TTLIM   010734          3191*   3194    3196*   3228    3252    3278    3282#
TTLOAD  010344          3188#   3194    3196
TT1     010334          3186#
TT2     010402          3193    3195    3197#
TT3     010546          3233#   3253
TT4     010620          3243    3250#
TT5     010640          3257#   3279
TT6     010716          3269    3276#
TT7     010736          3280    3284#
TVADFL  044702          9293#   9305*   9332#   9350
TVADHI  044706          9339#   9349*   9372#
TVADLO  044704          9338#   9348*   9373*   9374
TYPDS = 104404          8434    8896#   9270
TYPE  = 104400          2193    5290    5306    8432    8435    8540    8548    8648    8745    8821    8892#   8952    9079
                        9086    9093    9100    9107    9157    9170    9232    9247    9249    9253    9255    9279    9299
                        9314    9320    9378
TYPDC = 104401          8893#   9239    9265
TYPDN = 104403          8895#
TYPDS = 104402          8894#   9286
TYPVAD  044710          9294    9306    9346#
UBCLR   051442         10256   10295   10299#
```

DEKBDC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
UBERA   037116          7973#   8044
UBERA1  037252          7980*   8003#
UBERA2  037254          7983*   8004#
UBERA3  037250          7971*   8002#   8006*
UBERSS  033764          7271#
UBEBA   003730          2134#  10225*
UBEBB   037272          8011#
UBECC   003726          2133#  10224*
UBECCC  037274          7997*   8012#
UBECLR  003734          2136#  10283*  10299*
UBECR   033740          7258#   7285    8040*   8050*
UBECR1  003732          2135#   9602*   9603   10233*  10249   10253   10285   10292   10300
UBECR2  003736          2137#  10226*  10254   10293
UBECYL  050766         10197#
UBEDA1  050746         10189#  10211*
UBEDA2  050750         10190#  10212*  10227
UBEDB   003724          2132#   9467   10227*
UBEDD   037276          7981*   8013#
UBEDFL  045444          9413*   9484#   9605*
UBEEE   037300          7984*   8014#
UBEER1  050726          8046    9407*  10181#  10217*  10255*
UBEER2  050730          8051   10182#  10218*  10253*  10292*
UBEER3  050732          8052   10183#  10219*  10254*  10293*  10294*
UBEER4  050734         10184#
UBEFF   037320          8016    8022#
UBEFLG  050724          9402*  10180#  10199   10232*  10240*  10244*
UBEFT   040612          8320#
UBEFUN  050742         10187#  10210*  10231
UBEGG   037332          8027#
UBEHAN  050770          8917   10199#
UBEHH   037334          7996*   8028#
UBEH1   051002         10200   10204#
UBEH2   051106         10224#
UBEH3   051200         10235   10239#
UBEH4   051214         10229   10244#
UBEH5   051230         10239   10245   10248#
UBEH51  051232         10249#  10261
UBEH6   051274         10251   10260#
UBEII   037336          7982*   8029#
UBEJJ   037340          7985*   8030#
UBEKK   037360          8032    8038#
UBELL   037374          8041    8044#
UBEMA1  050752         10191#  10213*  10225   10265   10278*
UBEMA2  050754         10192#  10214*  10226   10266   10277*
UBERB   033776          7277#   7975    7977
UBERDY  051366         10220   10283#
UBEREG  003722          2131#   9469
UBESEC  050764         10196#
UBESUN  033752          7264#
UBESI   051304         10221   10265#
UBETMP  050740         10186#  10205*  10208
UBETRY  050762         10195#
UBEUNI  050744         10188#
UBEUSE  050736         10185#
UBEV    003750          2144#  10228
UBEVEC  050760         10194#  10216*  10246
```

```
UBEWCT  050756          10193#  10215*  10224   10279*  10280*
UBEYY   037400           8018    8022    8034    8038    8046#
UBEZZ   037440           8047    8057#
UDPAR0= 177660           1390#
UDPAR1= 177662           1391#
UDPAR2= 177664           1392#
UDPAR3= 177666           1393#
UDPAR4= 177670           1394#
UDPAR5= 177672           1395#
UDPAR6= 177674           1396#
UDPAR7= 177676           1397#
UDPDR0= 177620           1368#
UDPDR1= 177622           1369#
UDPDR2= 177624           1370#
UDPDR3= 177626           1371#
UDPDR4= 177630           1372#
UDPDR5= 177632           1373#
UDPDR6= 177634           1374#
UDPDR7= 177636           1375#
UIPAR0= 177640           1379#
UIPAR1= 177642           1380#
UIPAR2= 177644           1381#
UIPAR3= 177646           1382#
UIPAR4= 177650           1383#
UIPAR5= 177652           1384#
UIPAR6= 177654           1385#
UIPAR7= 177656           1386#
UIPDR0= 177600           1357#
UIPDR1= 177602           1358#
UIPDR2= 177604           1359#
UIPDR3= 177606           1360#
UIPDR4= 177610           1361#
UIPDR5= 177612           1362#
UIPDR6= 177614           1363#
UIPDR7= 177616           1364#
USESTK= 000600           1198#
USP    =%000006          1233#
UU     = 000012          4094#   4364
UUADR1  015344           4147*   4148*   4154    4155    4182*   4183*   4206*   4207*   4220*   4221*   4226    4227    4254*
                         4255*   4278*   4279*   4299#
UUADR2  015350           4145*   4146*   4156    4158    4208*   4218*   4219*   4228    4230    4280*   4301#
UUADR3  015354           4154*   4155*   4156*   4157*   4158*   4166    4168    4186    4203    4204    4226*   4227*   4228*
                         4229*   4230*   4238    4240    4258    4274    4275    4303#   4319    4320
UUDONE  015652           4286    4351    4356    4359#
UUERR1  015366           4150    4310#   4337
UUERR2  015402           4311    4314#
UUERR3  015522           4327    4333#
UUERR4  015524           4332    4334#
UUFLG1  015336           4108*   4202    4273    4285    4287*   4292#   4317
UUFLG2  015340           4112*   4151*   4212*   4223*   4284*   4295#   4339    4342    4345    4348    4355
UUFLG3  015342           4101*   4297#   4350    4352*
UUGM    015362           4110*   4118    4214    4289*   4307#
UUGS    015360           4109*   4116    4122    4288*   4306#
UUHIAD  014276           4106#
UULOAD  014274           4105#   4166    4168    4238    4240
UUTMP   015364           4308#   4326*   4336
```

```
UU0     014242          4095*   4113
UU1     014320          4112*   4290    4357
UU10    015100          4235*
UU11    015136          4248    4254*
UU12    015152          4245    4250    4258*
UU13    015244          4271    4278*   4347
UU14    015302          4285*   4353
UU2     014334          4116*   4120
UU3     014534          4152*   4184    4211
UU4     014570          4163*
UU5     014626          4176    4182*
UU6     014642          4173    4178    4186*
UU7     014736          4201    4206*   4341
UU8     015010          4218*   4344
UU9     015044          4224*   4256    4283
VV    = 000013          4070    4388*
VVADR1  016772          4443*   4444*   4451    4452    4479*   4480*   4504*   4505*   4518*   4519*   4524    4525    4552*
                        4553*   4576*   4577*   4594*
VVADR2  016776          4441*   4442*   4453    4455    4506*   4516*   45[illegible]*   4526    4528    4578*   4596*
VVADR3  017002          4451*   4452*   4453*   4454*   4455*   4463    4[illegible]    4483    4500    4501    4524*   4525*   4526*
                        4527*   4528*   4536    4538    4556    4572    4573    4598*   4615    4616
VVDONE  017300          4584    4646    4650    4653*
VVERR1  017014          4446    4606*   4633
VVERR2  017030          4607    4610*
VVERR3  017150          4623    4629*
VVERR4  017152          4628    4630*
VVFLG1  016764          4404*   4499    4571    4583    4587*   4590*   4613
VVFLG2  016766          4408*   4447*   4510*   4521*   4582*   4591*   4634    4637    4640    4643    4649
VVFLG3  016770          4396*   4592*   4645    4647*
VVGM    017010          4406*   4414    4512    4585*   4602*
VVGS    017006          4405*   4412    4418    4586*   4601*
VVHIAD  015720          4402*
VV_OAD  015716          4401*   4463    4465    4536    4538
VVTMP   017012          4604*   4622*   4632
VV0     015664          4389*   4409
VV1     015742          4408*   4588    4651
VV10    016526          4533*
VV11    016564          4546    4552*
VV12    016600          4543    4548    4556*
VV13    016672          4569    4576*   4642
VV14    016730          4583*   4648
VV2     015756          4412*   4416
VV3     016160          4449*   4481    4509
VV4     016214          4460*
VV5     016252          4473    4479*
VV6     016266          4470    4475    4483*
VV7     016362          4497    4504*   4636
VV8     016434          4516*   4639
VV9     016472          4522*   4554    4581
WAITLP  034000          7194    7282*   7284    7286    7288    7290    7292
WRRAND= 104420          7353    7521    7685    7854    8001    8911*
X     = 000001          2284*   2709
XADR1   005320          2324*   2325*   2330    2332    2348    2349    2469*   2470*   2475*
XADR2   005324          2348*   2349*   2350*   2351*   2352*   2360    2362    2393    2394    2427    2429    2477*   12381
                        12384
XADR3   005330          2322*   2323*   2330    2332    2350    2352    2459    2461    2463*   2465*   2466*   2471*   2472*
```

```
                    2479#
XADR4   005334      2481#
XDONE   005340      2367    2370    2483#
XHIADR  004572      2316#
XLOADR  004570      2315#   2317#   2360    2362
XX    = 000002      2509#   2909
XXADR1  006220      2547#   2548#   2553#   2554#   2555#   2563    2564    2695#
XXADR2  006224      2563#   2564#   2565#   2566#   2570    2572    2589    2591    2623    2624    2655    2657    2697#
                   12387   12390
XXCNST  006234      2570    2572    2702#
XXHIA   005500      2540#
XXLOA   005476      2539#   2541#   2589    2591
XXMASK  006230      2557#   2558#   2565    2566    2684#   2685#   2699#
XX1     005516      2547#
XX10    006156      2582    2599    2650    2674    2684#
XX11    006240      2692    2705#
XX2     005536      2552#   2693
XX3     005556      2550    2557#
XX4     005600      2561    2563#   2688
XX5     005666      2577    2580    2586#
XX6     005724      2596    2601    2605#
XX7     006046      2644    2652#
XX8     006104      2662    2671#
XX9     006120      2665    2667    2675#
X1      004632      2327#   2473
X10     005262      2460    2462    2465#
X11     005274      2337    2342    2469#
X12     005314      2464    2467    2473#
X2      004670      2340    2346#
X3      004724      2357#
X4      004762      2372    2375#
X5      005106      2418#
X6      005114      2414    2424#
X7      005152      2434    2443#
X8      005170      2437    2439    2448#
X9      005226      2422    2446    2457#
ZADHI   011062      3341#
ZADLO   011060      3340#   3409    3411    3437    3439
ZCMTBL  012002      3368    3376    3384    3546#
ZFLG1   011656      3343#   3466    3486    3509    3511#   3517#
ZFLG2   011660      3352#   3361    3371    3379    3502#   3514#   3519#
ZGM     011662      3350#   3390    3513#   3525#
ZGS     011664      3349#   3392    3512#   3526#
ZTABLE  011700      3366#   3367#   3374#   3375#   3382#   3383#   3400    3423    3536#   3546    3560    3562    3567
ZTABOT  012000      3495    3499    3537#
ZTHR    011670      3389    3528#   3568
ZTMP1   011674      3531#   3551#   3558    3559
ZTMP2   011676      3532#   3552#
Z1      011110      3361#   3503    3515
Z10     011366      3442    3445    3451#
Z11     011454      3465    3471#
Z12     011554      3430    3485    3493#
Z13     011566      3447    3498#
Z14     011610      3387    3419    3507#
Z15     012110      3510    3576#
Z2      011136      3362    3371#
```

```
23      011166          3372    3379#
24      011216          3380    3387#
25      011222          3369    3377    3385    3389#
27      011266          3404#   3500
28      011326          3429#   3496
29      011332          3432#
$AUTOB  001534          1641#
$BDADR  001522          1636#
$BDDAT  001526          1638#
$BELL   001706          1698#   8540    8567
$CHARC  042302          8650#   8660#   8667    8676#   8681#
$CKSWR= ****** U        8899
$CMTAG  001500          1624#   2150    2151    2159    2165    2166    2167
$CM1    = 000024        1656#   1657#   1658#   1659#   1660#   1661#   1662#   1663#   1664#   1665#   1666#   1667#   1668#
                        1669#   1670#   1671#   1672#   1673#   1674#   1675#   1676#
$CM2    = 000050        1656#   1657#   1658#   1659#   1660#   1661#   1662#   1663#   1664#   1665#   1666#   1667#   1668#
                        1669#   1670#   1671#   1672#   1673#   1674#   1675#   1676#
$CM3    = 000024        1654#   1656
$CM4    = 000024        1676#   1677#   1678#   1679#   1680#   1681#   1682#   1683#   1684#   1685#   1686#   1687#   1688#
                        1689#   1690#   1691#   1692#   1693#   1694#   1695#   1696#
$CRLF   001713          1700#   8548    8567    8649    8684    9233    9250    9256    12352   12358   12362   12368   12374
$DBLK   042750          8787    8821    8829#
$DB20   043332          8972#   9276    9312    9375
$DOAGN  041310          8428    8437    8441    8447#
$DTBL   042740          8790    8825#
$ENDAD  041300          1612    2191    8443#   8560
$ENDCT  041230          2165    8430#
$ENDMG  041317          8432    8451#
$ENULL  041314          8435    8450#
$EOP    041174          8187    8420#
$EOPCT  041222          2165*   8427#   8431
$ERFLG  001503          1627#   8460    8490    8492    8498#   8519    8535#   8567
$ERMAX  001515          1633#   2168#   8492    8514#   8519
$ERROR  041612          2159    8534#
$ERRPC  001516          1634#   8542#   8543#   8544    8567    9238    12310   12321   12336   12343   12348   12381   12384
                        12387   12390   12393   12396   12399   12402   12405   12408   12411   12416   12423   12428   12434
                        12451   12468   12471   12476
$ERRTB  001716          1717#   9244
$ERTTL  001512          1631#   8541#   8567
$ESCAP  001704          1697#   2167#   8513#   8556    8558    8567
$FILLC  001556          1652#   8653    8684
$FILLS  001555          1651#   8684
$GDADR  001520          1635#
$GDDAT  001524          1637#
$GET42  041252          8436#
$GTSWR= ****** U        8898
$HD     = 000003        1173    1174
$HINUM  043056          7316#   7319    7484#   7487    7652#   7655    7817#   7820    7988#   7991    8088#   8099    8148#
                        8151    8847    8855    8860#   8865#
$ICNT   001504          1628#   8505#   8506    8508#   8518
$ILLUP  043324          8922    8938    8957#
$INTAG  001535          1642#
$ITEMB  001514          1632#   3700#   5713#   8544#   8567    9035#   9236
$LF     001714          1701#   8567    8684
$LONUM  043060          7317#   7320    7485#   7488    7653#   7656    7818#   7821    7989#   7992    8089#   8147#   8150
                        8846    8853    8859#   8866#
```

```
$LPADR  001506          1629#  2169#  8496#  8511#  8516   8518
$LPERR  001510          1630#  2170#  2375#  2561#  2811#  2966#  5365#  5397#  5413#  5429#  5445#  5453#  5546#
                        5564#  5572#  5594#  5648#  5662#  5688#  5825#  5875#  5979#  6029#  6129#  6208#  6409#
                        6736#  6804#  8496   8512#  8518   8555
$MAIL = ****** U        2187   8511   8550   8637
$MXCNT  041610          8509   8518#
$NULL   001554          1650#  8655   8684
$NWTST= 000001          2224#  2226   2485#  2487   2707#  2709   2907#  2909   3096#  3098   3169#  3171   3287#
                        3289   3579#  3581   3790#  3792   4068#  4070   4362#  4364   4656#  4658   4793#  4795
                        4927#  4929   5088#  5090   5249#  5251   5330#  5332   5514#  5516   5630#  5632   5797#
                        5799   5951#  5953   6106#  6108   6344#  6346   6694#  6696   7106#  7108   8176#  8178
$OCNT   042530          8717#  8746#  8759#
$OCTVL  043434          8974   8999#
$OMODE  042532          8712#  8716#  8721   8724#  8735#  8761#
$OVER   041574          8471   8489   8497   8507   8515#
$PASS   001500          1625#  8424#  8425#  8433   8450   8503   8519
$PWRAD  043320          8955#
$PWRDN  043160          2163   8922#  8950
$PWRMG  043314          8953#
$PWRUP  043232          8932   8938#
$QUES   001712          1699#  8567   8684
$RAND   042760          7318   7486   7654   7819   7990   8098   8149   8842#
$RDCHR= ****** U        8899
$RDOEC= ****** U        8899
$RDLIN= ****** U        8899
$RDOCT= ****** U        8899
$REGAD  001560          1654#
$REG0   001562          1656#
$REG1   001564          1657#
$REG10  001602          1664#
$REG11  001604          1665#
$REG12  001606          1666#
$REG13  001610          1667#
$REG14  001612          1668#
$REG15  001614          1669#
$REG16  001616          1670#
$REG17  001620          1671#
$REG2   001566          1658#
$REG20  001622          1672#
$REG21  001624          1673#
$REG22  001626          1674#
$REG23  001630          1675#
$REG3   001570          1659#
$REG4   001572          1660#
$REG5   001574          1661#
$REG6   001576          1662#
$REG7   001600          1663#
$RESRE  042030          8601#  8900
$RTNAD  041312          8449#
$R2A  = ****** U        8901
$SAVRE  041772          8585#  8899
$SAVR6  043330          8931#  8939   8940#  8941#  8959#
$SCOPE  041334          2157   8469#
$SETUP= 000037          1577#  2156   2157   2159   2161   2163   2165   2166   2167   2169   2191   2194   8422
                        8470   8535   8553   8560
$STUP = 177777          1577#
```

```
$SVLAD  041546          8479    8510#
$SVPC = 000204          1610#   1615
$SWR  = 167400          1173    1174#   1175#   1182    1183    1184    1185    1186    1187    1188    1189    1696    1697
                        1698    2166    2167    2169    2170    2283    2508    2721    2921    3106    3178    3310    3590
                        3828    4093    4387    4669    4802    4940    5097    5273    5338    5522    5638    5815    5969
                        6114    6367    6709    7165    8186    8417    8423    8438    8448    8450    8461    8462    8463
                        8464    8465    8470    8482    8484    8485    8490    8491    8492    8499    8500    8501    8512
                        8515    8518    8526    8527    8528    8529    8530    8538    8545    8550    8553    8567    8956
$SWRMK= 000200          1176#   1189    1190    8465    8466    8486    8487
$TAB    052460          9321    10471#  12331   12358   12368   12377
$TIMES  001702          1696#   2166*   2283*   2508*   2721*   2921*   3106*   3178*   3310*   3590*   3828*   4093*   4387*
                        4669*   4802*   4940*   5097*   5338*   5522*   5638*   5815*   5969*   6367*   6709*   8423*   8499*
                        8506    8509*   8518
$TKB    001546          1647#   2219*   9151    9175*
$TKS    001544          1646#   2220*   9176*
$TMP0   001632          1676#   2288*   2513*   2726*   2926*   3111*   3183*   3314*   3596*   3833*   4100*   4394*   4674*
                        4807*   4945*   5102*   5277*   5343*   5527*   5643*   5820*   5974*   6119*   6372*   6714*   7169*
                        8190*   12310   12313   12321   12331   12336   12340   12343   12346   12348   12352   12362   12372
                        12381   12384   12387   12390   12393   12396   12399   12402   12405   12408   12411   12416   12419
                        12423   12428   12434   12443   12451   12460   12468   12471   12476   12493   12496   12501
$TMP1   001634          1677#   2448*   2675*   2871*   3065*   3466*   3486*   3701*   3944*   5387*   5393*   5403*   5409*
                        5419*   5425*   5435*   5441*   5553*   5577*   5584*   5599*   5608*   5671*   5695*   5714*   5758*
                        5786*   5841*   5887*   5995*   6041*   6161*   6323*   6330    6427*   6442*   6456*   6471*   6530*
                        6550*   6569*   6589*   6684*   6755*   6770*   6784*   6799*   6830*   6847*   6866*   6883*   6902*
                        6919*   6938*   6955*   6982*   7005*   7028*   7051*   7073*   7077*   7086*   7091*   7450*   7618*
                        7782*   7951*   8051*   8213*   8253*   8276*   8299*   9004*   9021*   9034*   12310   12313   12331
                        12336   12340   12343   12346   12348   12352   12362   12372   12384   12390   12393   12396   12402
                        12408   12419   12493   12496   12501
$TMP10  001652          1684#   3758*   4322*   4618*   4753*   4784*   4888*   4919*   5049*   5080*   5209*   5240*   6400*
                        6401*   6733*   6734*   12355   12365   12374   12437   12443   12454   12460
$TMP11  001654          1685#   4754*   4785*   4889*   4920*   5050*   5081*   5210*   5241*   12437   12454
$TMP12  001656          1686#   4755*   4775*   4890*   4910*   5051*   5071*   5211*   5231*   12434   12443   12451   12460
$TMP13  001660          1687#   4756*   4891*   5052*   5212*
$TMP14  001662          1688#   4750*   4751*   4885*   4886*   5046*   5047*   5206*   5207*   12437   12454
$TMP15  001664          1689#   4752*   4887*   5048*   5208*
$TMP16  001666          1690#   12434   12451
$TMP17  001670          1691#   4742*   4877*   5038*   5198*
$TMP2   001636          1678#   3138*   3154*   3246*   3271*   3467*   3487*   3691*   3754*   3945*   3946*   3968*   3986*
                        4202*   4273*   4315*   4499*   4571*   4611*   4777*   4912*   5073*   5233*   5311*   5388*   5389*
                        5394*   5404*   5405*   5410*   5420*   5421*   5426*   5436*   5437*   5442*   5451*   5457*   5554*
                        5578*   5585*   5600*   5609*   5672*   5673*   5696*   5715*   5760*   5840*   5869*   5886*   5915*
                        5994*   6023*   6040*   6069*   6162*   6163*   6324*   6426*   6440*   6441*   6455*   6469*   6470*
                        6529*   6548*   6549*   6568*   6587*   6588*   6683*   6754*   6768*   6769*   6783*   6797*   6798*
                        6829*   6846*   6865*   6882*   6901*   6918*   6937*   6954*   6980*   6981*   7003*   7004*   7026*
                        7027*   7049*   7050*   7080*   7084    7095*   7099    7452*   7620*   7784*   7953*   8052*   8214*
                        8254*   8277*   8300*   9005*   9022*   12310   12313   12321   12331   12340   12348   12352   12362
                        12372   12393   12396   12411   12416   12423   12428   12446   12463   12468   12471   12476   12493
                        12496   12501
$TMP20  001672          1692#   4744*   4879*   5040*   5200*
$TMP21  001674          1693#
$TMP22  001676          1694#
$TMP23  001700          1695#
$TMP3   001640          1679#   2450*   2677*   2872*   3066*   3139*   3155*   3247*   3272*   3468*   3488*   3753*   3943*
                        3969*   3987*   4203*   4274*   4317*   4500*   4572*   4613*   4745*   4778*   4880*   4913*   5041*
                        5074*   5201*   5234*   5586*   5610*   5716*   5761*   5787*   5870*   5916*   6024*   6070*   6325*
                        6666*   6671*   6675*   6680*   7081*   7096*   7451*   7619*   7783*   7952*   8215*   8255*   8278*
```

```
                    8301*  9006*  9023*  12331  12336  12340  12343  12348  12352  12362  12372  12384  12390
                   12402  12408  12411  12423  12428  12434  12451  12471  12476  12493  12501
$TMP4   001642      1680*   245*  2678*  2873*  3067*  3140*  3156*  3248*  3273*  3489*  3756*  3970*  3988*
                    4204*  4275*  4318*  4325*  4330   4501*  4573*  4614*  4621*  4626   4746*  4779*  4780*
                    4881*  4914*  4915*  5042*  5075*  5076*  5202*  5235*  5236*  5587*  5611*  5717*  5763*
                    5788*  5871*  5917*  6025*  6071*  6326*  6665*  6669*  6670*  6674*  6678*  6679*  7074*
                    7078*  7087*  7088*  7092*  7093*  9020*  12313  12331  12336  12340  12343  12348  12352
                   12362  12372  12396  12428  12446  12463
$TMP5   001644      1681*  3136*  3153*  3244*  3270*  3490*  3759*  4319*  4615*  4747*  4781*  4882*  4916*
                    5043*  5077*  5203*  5237*  5764*  5765*  5789*  6328*  6396*  6729*  12313  12336  12355
                   12365  12374  12411  12428  12434  12451
$TMP6   001646      1682*  3760*  4320*  4616*  4741*  4748*  4782*  4876*  4883*  4917*  5037*  5044*  5078*
                    5197*  5204*  5238*  5766*  6329*  6397*  6398*  6730*  6731*  12313  12331  12355  12365
                   12374  12437  12443  12454  12460
$TMP7   001650      1683*  3757*  4321*  4617*  4749*  4783*  4884*  4918*  5045*  5079*  5205*  5239*  5762*
                    6399*  6732*  12331  12355  12365  12374  12428  12446  12463
$TN   = 000033      1173*  2224   2283*  2284   2285   2485   2508*  2509   251C   2707   2721*  2722   2723
                    2907   2921*  2922   2923   3096   3106*  3107   3108   3169   3178*  3179   3180   3287
                    3310*  3311   3579   3590*  3591   3592   3790   3828*  3829   3830   4068   4093*  4094
                    4095   4362   4387*  4388   4389   4656   4669*  4670   4671   4793   4802*  4803   4804
                    4927   4940*  4941   4942   5088   5097*  5098   5099   5249   5273*  5274   5330   5338*
                    5339   5514   5522*  5523   5630   5638*  5639   5797   5815*  5816   5817   5951   5969*
                    5970   5971   6106   6114*  6344   6367*  6368   6694   6709*  6710   7106   7165*  8176
                    8186*
$TPB    001552      1649*  8673*  8684
$TPFLG  001557      1653*  8631   8684
$TPS    001550      1648*  8671   8684
$TRAP   043062      2161   8876*
$TRP  = 000026      8884*  8893*  8894*  8895*  8896*  8897*  8899   8900*  8901*  8902   8903*  8904*  8905*
                    8906*  8907*  8908*  8909*  8910*  8911*  8912*  8913   8914*  8915*  8916*  8917*  8918*
$TRPAD  043104      8881   8891*
$TSTNM  001502      1626*  2148*  2288   2513   2726   2926   3111   3183   3314   3596   3833   4100   4394
                    4674   4807   4945   5102   5277   5343   5527   5643   5820   5974   6119   6372   6714
                    7169   8190   8422*  8460   8488   8510*  8515   8519   8537   8567
$TYPBN= ****** U    8897
$TYPDS  042534      8775*  8896
$TYPE   042066      8631*  8884   8892
$TYPEC  042236      8652   8659   8666   8671*  8672
$TYPEX  042304      8677   8679   8682*
$TYPOC  042332      8715*  8893
$TYPON  042346      8714   8717*  8895
$TYPOS  042306      8710*  8894
$XTSTR  041344      8473*
$SGET4= 000001      8438*  8440*
$OFILL  042531      8711*  8715*  8725   8760*
$40CAT= ****** U    8470   8547
.     = 073450      1596*  1600*  1610   1611*  1613*  1615*  1623*  1702   2154   2169   2170   2196*  3536*
                    5292   5296   5854   5857*  5903   5906*  6008   6011*  6057   6060*  7069*  8321*  8450
                    8454   8518   8519   8567   8684   8829*  8934   8958   8999*  9485*  12306*
```

```
ADDDPA  1151#  2346   2782   3554   3879   4152   4224   4449   4522
ADJUST  1137#  5853   5902   6007   6056
BYTLT1  1140#  4734   4869   5030   5190
BYTLT2  1139#  4774   4909   5070   5230
CHAIN   1120#  8438
CLRMAC  1142#  9046
CLRRFL  1136#
CMPDPA  1150#  2327   2357   2424   2568   2586   2652   2763   2793   2846   2972   2989   3040   3404   3432
        3654   3729   3892   4035   4163   4235   4460   4533
CNVDPA  1149#  2376   2605
CNVMAP  1144#  2812   3008
COMMEN     1#  1576#
DD      1117#  7297   7465   7633   7798   7966
DRIVER  1116#  7300   7468   7636   7801
ENDCOM     1#  1576#
ERCLR   1119#  8564
ERROR   1199#  2421   2454   2649   2681   2842   2876   3037   3070   3141   3157   3249   3274   3469   3491
        3692   3761   3947   3971   3989   4205   4276   4323   4502   4574   4619   4757   4786   4892   4921
        5053   5082   5213   5242   5312   5390   5395   5406   5411   5422   5427   5438   5443   5452   5458
        5555   5579   5588   5601   5612   5674   5697   5767   5791   5842   5872   5888   5918   5996   6026
        6042   6072   6164   6332   6336   6338   6428   6443   6457   6472   6531   6551   6570   6590   6687
        6756   6771   6785   6800   6831   6848   6867   6884   6903   6920   6939   6956   6983   7006   7029
        7052   7083   7098   7453   7621   7785   7954   8053   8218   8222   8224   8257   8280   8303   8370
        8388   8406   9008   9025   9699   9857  10015  10201  10330
ESCAPE     1#  1576#
GETPRI     1#  1576#
GETSWR     1#  1576#  2194#
HANREG  1119#  9677   9835   9993  10179  10308
IIMAC1  1126#  5797   5951
ITEMAC  1138#  1853
JJM1    1157#  5382   5398   5414   5430
KKM1    1147#  6483   6487   6491   6495   6502   6506   6510   6514
KKM2    1148#  6519   6538   6558   6577
KMAC1   1138#
KMAC5   1131#
MMAC1   1126#
MMAC2   1127#
MMAC3   1128#
MMAC4   1129#
MMAC5   1130#
MMM1    1154#  6824   6841   6860   6877   6896   6913   6932   6949
MMM2    1155#  6971   6993   7016   7039
MSG     2224#  2226   2484#  2487   3096#  3098   3286#  3289   3578#  3581   3790#  3792   4067#  4070   4361#
        4364   5248#  5251   5330#  5332   5514#  5516   5630#  5632   5797#  5799   5951#  5953   6105#  6108
        6344#  6346   6693#  6696   8175#  8178
MSGS    7105#  7108
MSG1    2706#  2709   2909
MSG2    3168#  3171
MSG201  5796#  5799   5953
MSG3    4067#  4070   4364   4655#  4658   4929
MSG300 11382#  11424  11466  11509
MSG301 11551#  11593  11633  11675
MSG4    4792#  4795   5090
MULT       1#  1576#
NEWTST     1#  1576#  2224   2485   2707   2907   3096   3169   3287   3579   3790   4068   4362   4656   4793
        4927   5088   5249   5330   5514   5630   5797   5951   6106   6344   6694   7106   8176
```

```
NMAC1     1132#
NMAC2     1133#
NMAC3     1135#
POP          1#   1576#  8606   8816   8861   8943   8944
PUSH         1#   1576#  8586   8775   8842   8924   8930
QQM1      1156#   6241   6261   6281
REPORT       1#   1576#
SCOPE     1200#   2282   2507   2720   2920   3105   3177   3309   3589   3827   4092   4386   4668   4801   4939
          5096    5272   5337   5521   5637   5814   5968   6113   6366   6708   7164   8185   8421
SETMAP    1143#   4948   5108
SETMM     1146#   2731   2929   3197   3322   3629   3836   4126   4422   7170
SETPRI       1#   1576#
SETTRA    8884#   8893   8894   8895   8896   8899   8900   8902   8903   8904   8905   8906   8907   8908   8909
          8910    8911   8913   8914   8915   8916   8917
SETUP        1#   1576#  2149
SKBREX    1136#   5822   5976
SKIP         1#   1576#
SLASH        1#   1576#
SPACE     1576#
SSKAD     1151#   2285   2510   2723   2923   3108   3180   3311   3592   3830   4095   4389   4671   4804   4942
          5099    5274   5339   5523   5639   5817   5971   6114   6368   6710   7165   8186
STARS        1#   1576#  1608   1619   1702   2224   2281   2485   2506   2707   2719   2907   2919   3096   3104
          3169    3176   3287   3308   3579   3588   3790   3826   4068   4091   4362   4385   4656   4667   4793
          4800    4927   4938   5088   5095   5249   5271   5330   5336   5514   5520   5630   5636   5797   5813
          5951    5967   6106   6112   6344   6365   6694   6707   7106   7163   8176   8184   8413   8457   8522
          8570    8616   8687   8765   8833   8870   8920   8936   8963   9614   9672
SWRSU        1#   1576#  2171#
THIT      1152#   5835   5881   5989   6035   6421   6435   6450   6464   6524   6543   6563   6582   6749   6763
          6778    6792   6975   6998   7021   7044
TOMAPO    1141#   3218   3233   3257
TOPAR3    1145#   3451   3471   3674   4187   4259   4484   4557
TRMTRP    8884#
TSTHIT    1153#   6419   6432   6447   6461   6747   6760   6775   6789   6973   6996   7019   7042
TYPBIN       1#   1576#
TYPDEC       1#   1576#  8433
TYPNAM       1#   1576#  2187
TYPNUM       1#   1576#
TYPOCS       1#   1576#
TYPOCT       1#   1576#
TYPTXT       1#   1576#
UMAC1     1121#
UMAC2     1122#   5924   5934   6078   6088
UMAC3     1124#   5867   5913   6021   6067
$$CMRE    1617#   1656   1657   1658   1659   1660   1661   1662   1663   1664   1665   1666   1667   1668   1669
          1670    1671   1672   1673   1674   1675
$$CMTM    1617#   1676   1677   1678   1679   1680   1681   1682   1683   1684   1685   1686   1687   1688   1689
          1690    1691   1692   1693   1694   1695
$$ESCA       1#   1576#
$$NEWT       1#   1576#  2224   2485   2707   2907   3096   3169   3287   3579   3790   4068   4362   4656   4793
          4927    5088   5249   5330   5514   5630   5797   5951   6106   6344   6694   7106   8176
$$SET     8884#   8893   8894   8895   8896   8899   8900   8902   8903   8904   8905   8906   8907   8908   8909
          8910    8911   8913   8914   8915   8916   8917
$$SKIP       1#   1576#
.EQUAT       1#
.HEADE       1#   1160#  1163
.KT11        1#
```

```
.SETUP      1#   1160#   1577
.SWRHI      1#   1160#   1178
.SWRLO   1190#
.$ACT1      1#   1160#   1606
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#   1160#   1594
.$CMTA      1#   1160#   1617
.$DB2D      1#
.$DB20      1#   1162#   8961
.$DIV       1#
.$EOP       1#   1161#   8411
.$ERRO      1#   1161#   8520
.$ERRT      1#
.$MULT      1#
.$POWE      1#   1162#   8918
.$RAND      1#   1161#   8831
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2A2      1#
.$SAVE      1#   1161#   8568
.$SB2D      1#
.$SB20      1#
.$SCOP      1#   1161#   8455
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#   1162#   8868
.$TYPB      1#
.$TYPD      1#   1162#   8763
.$TYPE      1#   1161#   8614
.$TYPO      1#   1161#   8685
.$40CA      1#
.1170       1#   1160#   1192
```

```
ADC     2351  2787  3561  3695  3774  3883  3992  4157  4207  4229  4279  4454  4505  4527  4577
        8101  8122  8854  8857  9370 10132 10274
ADD     2350  2352  2400  2630  2786  2788  3227  3251  3277  3459  3479  3493  3498  3560  3562
        3570  3681  3694  3882  3884  3922  3946  3991  4156  4158  4194  4206  4208  4228  4230
        4266  4278  4280  4453  4455  4491  4504  4506  4526  4528  4564  4576  4578  4692  4704
        4716  4751  4780  4886  4915  4962  4963  4988  5000  5012  5047  5076  5122  5123  5207
        5236  5349  5351  5353  5389  5405  5421  5437  5657  5673  5708  5946  6100  6149  6163
        6304  6392  6394  6398  6401  6441  6470  6549  6588  6670  6679  6725  6727  6731  6734
        6769  6798  6816  6818  6858  6894  6930  6967  6969  6981  7004  7027  7050  7088  7093
        8079  6097  8100  8121  8144  8158  8227  8641  8713  8783  8795  8853  8855  8856  8858
        8997  9244  9277  9363  9369  9376  9446  9502  9508  9703  9861 10020 10131 10206 10273
       10335
ASH     4958  5118  9243 10096 10112 10114
ASHC    2401  2631  3457  3477  3679  3924  4192  4264  4489  4562  7313  7481  7649  7814  7979
        8093  8125  9355  9367
ASL     2465  2469  2554  2684  2887  2891  2959  3073  7218  7226  7229  7232  8367  8386  8404
        8849  8880  9139  9215  9356  9538  9566  9593
ASLB    8800
ASR     8389  9940  9941
BCC     8801  9140
BEQ     2192  2687  3076  3135  3152  3243  3269  3430  3485  3699  4029  4210  4282  4508  4580
        4687  4698  4710  4722  4739  4821  4833  4845  4857  4874  4982  4994  5006  5018  5035
        5142  5154  5166  5178  5195  5316  5392  5408  5424  5440  5450  5456  5475  5582  5607
        5699  5705  5712  5743  5746  5750  5828  5926  5936  5945  5982  6080  6090  6095  6307
        6535  6555  6574  6594  6620  6835  6852  6871  6888  6907  6924  6943  6960  6989  7012
        7035  7058  7205  7213  7446  7614  7778  7947  8047  8161  8211  8232  8353  8362  8375
        8381  8393  8399  8437  8441  8485  8489  8491  8495  8504  8536  8539  8554  8557  8644
        8679  8740  8984  9015  9137  9246  9252  9325  9351  9353  9604  9698  9737  9756  9802
        9804  9806  9817  9856  9894  9914  9965  9967  9974 10014 10061 10159 10200 10235 10329
       10367 10398 10428 10430 10438
BGE     2692  3081  8507
BGT     6193  8428  8747  8809  8983
BHI     8493
BIC     2317  2399  2541  2565  2566  2629  2754  2948  2970  2971  3454  3474  3569  3618  3676
        3766  3769  3780  3855  3858  3921  3957  3982  4123  4189  4215  4261  4325  4329  4419
        4486  4513  4559  4621  4625  4680  4813  4974  5134  5347  5542  5559  5664  5677  5756
        5765  5784  6154  6167  6171  6173  6180  6182  6189  6191  6218  6220  6389  6723  6833
        6850  6869  6886  6905  6922  6941  6958  8096  8335  8342  8425  8487  8737  8996  9017
        9154  9602  9771  9782  9929  9947 10097 10113 10115 10116 10126 10267 10404
BICB    8363  8382  8400
BIS     2553  2958  3763  3764  5754  5782  8742  8743  8803  8804  9218  9537  9565  9592  9751
        9786  9909  9942  9950 10050 10051 10057 10117 10118 10138 10143 10226 10383 10412
BISB    2220  9176
BIT     3464  3484  3688  3748  3940  4200  4270  4310  4496  4568  4606  4735  4870  5031  5191
        5280  5315  5385  5401  5417  5433  5551  5575  5597  5669  5693  5827  5837  5883  5981
        5991  6037  6159  6330  6423  6437  6452  6466  6526  6545  6565  6584  6751  6765  6780
        6794  6827  6844  6863  6880  6899  6916  6935  6952  6977  7000  7023  7046  7212  8210
        8361  8380  8398  8470  8484  8494  8501  8538  8545  8553  9014  9136  9192  9352  9603
        9736  9755  9801  9803  9805  9807  9809  9816  9893  9913  9964  9966  9968  9973 10060
       10152 10158 10234 10366 10387 10427 10429 10431 10437
BITB    8667
BLE     3124  3195
BLO     2339  2369  2436  2579  2598  2664  2775  2805  2858  2983  3001  3052  3416  3444  3666
        3741  3903  4047  4175  4247  4472  4545
BLOS    3130  3145  3229  3253
BLT     6175  6184  6222  8658  8748  8792  8808
```

B07

BMI	8250	8273	8296	8799	9795	4959	10147	10155	10296	10422					
BNE	2154	2178	2190	2206	2331	2333	2361	2363	2423	2430	2460	2462	2571	2573	2590
	2592	2656	2658	2767	2769	2797	2799	2850	2852	2881	2883	2975	2977	2993	2995
	3044	3046	3122	3161	3193	3279	3362	3372	3390	3410	3412	3438	3440	3465	3496
	3500	3510	3658	3660	3689	3705	3733	3735	3749	3895	3897	3941	3996	4039	4041
	4059	4062	4167	4169	4201	4239	4241	4271	4286	4311	4340	4343	4346	4349	4351
	4356	4464	4466	4497	4537	4539	4569	4584	4607	4635	4638	4641	4644	4646	4650
	4691	4702	4714	4726	4736	4762	4766	4770	4825	4837	4849	4861	4871	4897	4901
	4905	4996	4998	5010	5022	5032	5058	5062	5066	5146	5159	5170	5182	5192	5218
	5222	5226	5281	5305	5386	5402	5418	5434	5462	5482	5488	5494	5552	5576	5598
	5620	5670	5694	5723	5838	5884	5929	5939	5992	6038	6083	6093	6160	6313	6318
	6331	6335	6424	6438	6453	6467	6480	6527	6546	6566	6585	6598	6607	6628	6635
	6642	6752	6766	6781	6795	6828	6845	6864	6881	6900	6917	6936	6953	6978	7001
	7024	7047	7225	7228	7231	7284	7286	7288	7290	7292	7438	7606	7770	7939	8041
	8217	8221	8471	8502	8546	8561	8638	8646	8654	8668	8675	8738	8797	8852	8942
	9075	9077	9084	9091	9098	9105	9156	9166	9193	9237	9259	9263	9268	9274	9284
	9291	9297	9304	9310	9444	9536	9564	9591	9808	9810	9969	10153	10432		
BPL	8551	8632	8672	8736	8783	8813	9754	9765	9767	9790	9797	9812	9829	9912	9923
	9925	9954	9961	9971	9986	10080	10089	10091	10141	10149	10157	10173	10251	10261	10288
	10301	10386	10398	10400	10416	10424	10434	10450							
BR	2180	2194	2464	2467	2550	2650	2674	2885	2899	2955	3038	3063	3162	3280	3369
	3377	3385	3696	3782	4184	4256	4332	4481	4554	4628	4689	4700	4712	4724	4823
	4835	4847	4859	4984	4996	5008	5020	5144	5156	5168	5180	5292	5296	5468	5852
	5898	5920	6006	6052	6074	6178	6187	6196	6333	6337	6499	6613	6667	6672	6676
	6681	7061	7075	7089	8473	8479	8482	8497	9500	8634	8651	8661	8670	8677	8714
	8729	8750	8794	8811	8934	8958	8998	9081	9088	9095	9102	9266	9271	9281	9289
	9295	9301	9307	9316	9326	9422	9433	9451	9461	9471					
CLC	9359														
CLR	2148	2152	2166	2167	2219	2301	2322	2323	2324	2457	2458	2471	2472	2528	2548
	2558	2699	2690	2738	2758	2759	2760	2878	2879	2893	2894	2936	2953	2963	3078
	3079	3140	3156	3205	3221	3236	3248	3260	3273	3329	3343	3352	3367	3375	3383
	3507	3508	3514	3548	3549	3604	3623	3624	3625	3636	3648	3649	3650	3843	3860
	3866	3871	4101	4108	4112	4133	4145	4146	4148	4183	4218	4219	4221	4255	4396
	4404	4408	4429	4442	4444	4480	4517	4519	4553	4682	4683	4693	4694	4705	4706
	4717	4718	4742	4749	4752	4756	4778	4781	4785	4877	4884	4887	4891	4913	4916
	4920	4953	4960	4977	4978	4989	4990	5001	5002	5013	5014	5038	5045	5048	5052
	5074	5077	5081	5113	5120	5198	5205	5208	5212	5234	5237	5241	5298	5299	5363
	5380	5460	5548	5666	5682	5683	5685	5686	5687	5689	5773	5823	5849	5871	5895
	5917	5977	6003	6025	6049	6071	6121	6377	6382	6417	6485	6489	6493	6497	6504
	6508	6512	6516	6522	6541	6561	6580	6603	6666	6671	6699	6745	6822	6825	6842
	6861	6878	6897	6914	6933	6950	7073	7086	7177	7193	7202	7203	7210	7312	7374
	7394	7414	7431	7449	7480	7542	7562	7582	7599	7617	7648	7706	7726	7746	7763
	7781	7813	7875	7895	7915	7932	7950	7978	8019	8035	8050	8089	8116	8153	8198
	8215	8351	8358	8378	8396	8422	8423	8499	8513	8565	8727	8786	8789	8940	8979
	9052	9053	9054	9055	9057	9058	9135	9153	9175	9212	9235	9305	9349	9354	9366
	9398	9399	9400	9401	9402	9403	9404	9405	9406	9407	9425	9436	9454	9464	9474
	9522	9550	9577	9714	9715	9716	9741	9745	9775	9778	9872	9873	9874	9899	9903
	9933	9935	10032	10033	10034	10046	10066	10071	10102	10104	10107	10135	10170	10217	10218
	10219	10240	10244	10277	10278	10283	10299	10346	10347	10348	10372	10376			
CLRB	4819	4831	4843	4855	5140	5152	5164	5176	8258	8259	8281	8282	8304	8305	8371
	8389	8407	8498	8650	8676	8815	9409	9410	9411	9412	9413	9523	9551	9578	
CMP	2153	2177	2191	2330	2332	2360	2362	2427	2429	2444	2453	2570	2572	2589	2591
	2655	2657	2672	2680	2691	2766	2768	2796	2798	2849	2851	2865	2967	2875	2974
	2976	2992	2994	3043	3045	3059	3061	3069	3080	3123	3129	3134	3144	3151	3160
	3194	3228	3242	3252	3268	3278	3371	3379	3409	3411	3429	3437	3439	3495	3499
	3657	3659	3732	3734	3783	3894	3896	4028	4038	4040	4056	4058	4061	4166	4169

```
        4238  4240  4333  4339  4342  4345  4348  4463  4465  4536  4538  4629  4634  4637  4640
        4643  4686  4697  4709  4721  4738  4761  4765  4769  4820  4824  4832  4836  4844  4848
        4856  4860  4873  4896  4900  4904  4981  4993  5005  5017  5034  5057  5061  5065  5141
        5145  5153  5157  5169  5177  5181  5194  5217  5221  5225  5391  5407  5423  5439  5449
        5455  5461  5481  5487  5493  5581  5606  5698  5745  5749  5759  5864  5910  5925  5928
        5931  5935  5938  5941  5944  6019  6064  6079  6082  6085  6089  6092  6095  6098  6174
        6183  6192  6221  6327  6534  6554  6573  6593  6606  6627  6634  6641  6834  6851  6870
        6887  6906  6923  6942  6959  6988  7011  7034  7057  7224  7227  7230  7239  8160  8216
        8220  8231  8480  8506  8560  8807  9007  9024  9028  9036  9110  9155  9165  9195  9221
        9424  9435  9443  9453  9463  9473
CMPB    5165  8488  8492  8643  8645  8653  8674  8678  9267  9273  9283  9290  9296  9303  9309
COM     3143  3150  3159  3250  3267  3276  3765  3767  3768  3770  5590  5605  5755  5757  5783
        5785  6214  6301  10279
COMB    6837  6854  6873  6890  6909  6926  6945  6962
DEC     2686  3075  4209  4281  4352  4507  4579  4647  5474  5479  5619  5722  6306  6312  6317
        6619  6625  7437  7605  7769  7938  8040  8426  8982  9242
DECB    8657  8660  8735  8746
DIV     8154  9776  9934  9936  10103 10105 10108
EMT     1199
HALT    1600  4064  8552  8562  8633  8933  8957  9174  9318  9700  9858  10016 10202 10331
INC     2189  2205  2552  2957  3502  3511  3773  4764  4768  4827  4839  4851  4899  4903  5060
        5064  5148  5160  5172  5220  5224  5318  5706  7217  7237  7358  7526  7690  7859  8006
        8230  8366  8385  8403  8424  8505  8541  8741  8749  8793  8851  8941  8985  9138  9323
        10280
INCB    8510  8535  8680  9539  9567  9594
IOT     1200
JMP     1604  2337  2340  2342  2367  2370  2372  2422  2434  2437  2439  2446  2473  2577  2580
        2582  2596  2599  2601  2662  2665  2667  2688  2693  2773  2776  2778  2803  2806  2808
        2843  2856  2859  2861  2869  2896  2981  2984  2986  2999  3002  3004  3050  3053  3055
        3077  3082  3387  3414  3417  3419  3442  3445  3447  3503  3515  3664  3667  3669  3709
        3739  3742  3744  3750  3786  3901  3904  3906  3948  3993  4001  4031  4045  4048  4050
        4060  4063  4173  4176  4178  4211  4245  4248  4250  4283  4290  4312  4341  4344  4347
        4353  4357  4470  4473  4475  4509  4543  4546  4548  4581  4588  4608  4636  4639  4642
        4648  4651  4728  4737  4740  4763  4767  4771  4772  4863  4872  4879  4898  4902  4906
        4907  5024  5033  5036  5059  5063  5067  5068  5184  5193  5196  5213  5223  5227  5228
        5282  5308  5313  5321  5476  5485  5491  5497  5621  5700  5724  5747  5751  5778  5793
        5829  5927  5933  5937  5947  5983  6081  6087  6091  6101  6223  6224  6257  6277  6297
        6308  6314  6319  6339  6600  6621  6632  6639  6646  7084  7099  7216  7221  7294  7442
        7610  7774  7943  8044  8196  8219  8223  8225  8235  8237  8260  8283  8306  8365  8384
        8402  8448  9029  9039  9059  9177  9197  9240  9260  9747  9905  10074 10246 10378
JSR     2416  2646  2837  3034  3368  3376  3384  3702  3964  3983  4688  4699  4711  4723  4822
        4834  4846  4858  4983  4995  5007  5019  5143  5155  5167  5179  5718  5826  5980  6536
        6556  6575  6595  6836  6853  6872  6889  6908  6925  6944  6961  6990  7013  7036  7059
        7189  7191  7309  7318  7373  7379  7393  7398  7413  7417  7430  7435  7477  7486  7541
        7547  7561  7566  7581  7585  7598  7603  7645  7654  7705  7711  7725  7730  7745  7749
        7762  7767  7810  7819  7874  7880  7894  7899  7914  7918  7931  7936  7974  7990  8018
        8022  8034  8038  8098  8149  8191  8202  8203  8439  8443  8547  8652  8659  8666  9037
        9276  9294  9306  9312  9375  9419  9421  9430  9432  9440  9447  9450  9458  9460  9468
        9470  9718  9719  9720  9722  9740  9746  9761  9823  9876  9877  9879  9880  9898  9904
        9919  9980  10036 10037 10039 10042 10065 10070 10084 10166 10220 10221 10239 10245 10256
        10295 10349 10350 10351 10371 10377 10393 10444
MOV     2151  2155  2157  2158  2159  2160  2161  2162  2163  2164  2165  2169  2170  2173  2174
        2175  2176  2181  2183  2184  2185  2208  2210  2211  2212  2213  2215  2217  2218  2221
        2222  2283  2286  2289  2295  2296  2297  2298  2299  2302  2303  2304  2305  2306  2307
        2308  2309  2310  2320  2325  2348  2349  2375  2393  2394  2398  2402  2404  2405  2407
        2408  2409  2411  2412  2414  2445  2448  2450  2451  2455  2463  2508  2511  2514  2522
```

2523 2524 2525 2526 2529 2530 2531 2532 2533 2534 2535 2536 2537 2544 2547
2549 2557 2560 2561 2563 2564 2623 2624 2628 2632 2634 2635 2637 2638 2639
2641 2642 2644 2673 2675 2677 2678 2682 2721 2724 2727 2728 2732 2733 2734
2735 2736 2739 2740 2741 2742 2743 2744 2745 2747 2748 2756 2761 2784 2785
2811 2815 2816 2820 2822 2824 2826 2828 2829 2830 2832 2833 2835 2868 2871
2872 2873 2877 2884 2921 2924 2927 2930 2931 2932 2933 2534 2937 2938 2939
2940 2941 2942 2943 2950 2952 2954 2962 2965 2966 2968 2969 3012 3013 3017
3019 3021 3023 3025 3026 3027 3029 3030 3032 3062 3065 3066 3067 3071 3106
3109 3112 3114 3120 3125 3127 3128 3132 3133 3136 3138 3139 3147 3148 3149
3153 3154 3155 3178 3181 3184 3186 3191 3196 3199 3200 3201 3202 3203 3206
3207 3208 3209 3210 3211 3212 3214 3215 3216 3220 3222 3223 3226 3231 3235
3237 3238 3241 3244 3246 3247 3255 3259 3261 3262 3265 3266 3270 3271 3272
3310 3312 3315 3323 3324 3325 3326 3327 3330 3331 3332 3333 3334 3335 3336
3349 3350 3366 3374 3382 3389 3390 3392 3397 3398 3400 3423 3453 3455 3456
3458 3466 3467 3468 3473 3475 3476 3478 3486 3487 3488 3489 3490 3512 3513
3546 3550 3551 3552 3558 3559 3566 3567 3568 3590 3594 3605 3606 3608 3609
3610 3611 3613 3617 3619 3621 3622 3626 3627 3630 3631 3632 3633 3634 3637
3638 3639 3640 3641 3642 3643 3645 3646 3652 3673 3675 3677 3678 3680 3691
3701 3706 3707 3708 3753 3754 3756 3757 3758 3759 3760 3772 3776 3779 3784
3785 3828 3831 3834 3837 3838 3839 3840 3841 3844 3845 3846 3847 3848 3849
3850 3857 3859 3862 3863 3865 3867 3868 3870 3874 3880 3881 3910 3916 3917
3920 3925 3927 3934 3935 3936 3943 3944 3945 3952 3955 3956 3968 3969 3970
3979 3981 3986 3987 3988 3997 3998 3999 4000 4024 4025 4030 4054 4093 4097
4099 4109 4110 4113 4114 4116 4118 4122 4124 4127 4128 4129 4130 4131 4134
4135 4136 4137 4138 4139 4140 4142 4143 4147 4149 4150 4151 4154 4155 4182
4186 4188 4190 4191 4193 4202 4203 4204 4212 4214 4216 4220 4222 4223 4226
4227 4254 4258 4260 4262 4263 4265 4273 4274 4275 4284 4287 4288 4289 4315
4317 4318 4319 4320 4321 4322 4326 4327 4328 4330 4334 4336 4337 4387 4391
4393 4405 4406 4409 4410 4412 4414 4418 4420 4423 4424 4425 4426 4427 4430
4431 4432 4433 4434 4435 4436 4438 4439 4441 4443 4445 4446 4447 4451 4452
4479 4483 4485 4487 4488 4490 4499 4500 4501 4510 4512 4514 4516 4518 4520
4521 4524 4525 4552 4556 4558 4560 4561 4563 4571 4572 4573 4582 4585 4586
4587 4611 4613 4614 4615 4616 4617 4618 4622 4623 4624 4626 4630 4632 4633
4669 4672 4675 4677 4679 4681 4741 4744 4745 4746 4747 4748 4750 4753 4754
4755 4758 4760 4775 4777 4779 4782 4783 4784 4802 4805 4808 4810 4812 4814
4816 4817 4828 4829 4840 4841 4852 4853 4876 4879 4880 4881 4882 4883 4885
4888 4889 4890 4893 4895 4910 4912 4914 4917 4918 4919 4940 4943 4947 4949
4951 4952 4954 4956 4957 4959 4961 4965 4966 4968 4969 4971 4973 4975 5037
5040 5041 5042 5043 5044 5046 5049 5050 5051 5054 5056 5071 5073 5075 5078
5079 5080 5097 5100 5106 5109 5111 5112 5114 5116 5117 5119 5121 5125 5126
5128 5129 5131 5133 5135 5137 5138 5149 5150 5161 5162 5173 5174 5197 5200
5201 5202 5203 5204 5206 5209 5210 5211 5214 5216 5231 5233 5235 5238 5239
5240 5275 5278 5284 5286 5267 5295 5297 5298 5300 5303 5311 5317 5328 5338
5340 5342 5344 5345 5346 5348 5350 5352 5354 5356 5361 5364 5365 5366 5367
5369 5370 5371 5372 5374 5375 5376 5377 5384 5387 5388 5393 5394 5397 5400
5403 5404 5409 5410 5413 5416 5419 5420 5425 5426 5429 5432 5435 5436 5441
5442 5445 5447 5448 5451 5453 5454 5457 5464 5465 5466 5467 5470 5471 5472
5477 5478 5483 5484 5489 5490 5495 5496 5522 5524 5526 5529 5530 5531 5534
5535 5536 5538 5541 5543 5544 5545 5546 5553 5554 5558 5560 5562 5563 5564
5566 5570 5571 5572 5574 5577 5578 5584 5585 5586 5587 5592 5593 5594 5596
5599 5600 5604 5608 5609 5610 5611 5617 5618 5638 5640 5642 5645 5646 5647
5648 5649 5650 5651 5653 5655 5660 5661 5662 5663 5665 5671 5672 5676 5678
5680 5681 5684 5688 5691 5692 5695 5696 5707 5714 5715 5716 5717 5720 5721
5740 5741 5744 5753 5758 5760 5761 5762 5763 5764 5766 5769 5776 5777 5780
5786 5787 5788 5789 5815 5818 5821 5825 5830 5831 5832 5840 5841 5847 5848
5850 5863 5865 5869 5870 5874 5875 5876 5878 5886 5887 5893 5894 5896 5909

5911	5915	5916	5932	5942	5969	5972	5975	5979	5984	5985	5986	5994	5995	6001
6002	6004	6017	6019	6023	6024	6028	6029	6030	6032	6040	6041	6047	6048	6050
6063	6065	6069	6070	6086	6096	6115	6118	6125	6129	6131	6132	6139	6143	6144
6145	6147	6151	6152	6153	6155	6156	6161	6162	6166	6168	6170	6172	6176	6177
6179	6181	6185	6186	6188	6190	6194	6195	6197	6198	6200	6201	6202	6204	6208
6212	6213	6217	6219	6249	6251	6253	6255	6269	6271	6273	6275	6289	6291	6293
6295	6310	6311	6316	6323	6324	6325	6326	6328	6329	6367	6369	6374	6376	6378
6379	6380	6388	6391	6393	6396	6397	6399	6400	6403	6406	6407	6409	6411	6414
6426	6427	6440	6442	6455	6456	6469	6471	6483	6484	6486	6487	6488	6490	6491
6492	6494	6495	6496	6498	6502	6503	6505	6506	6507	6509	6510	6511	6513	6514
6515	6517	6520	6521	6523	6529	6530	6539	6540	6542	6548	6550	6559	6560	6562
6568	6569	6578	6579	6581	6587	6589	6599	6604	6609	6610	6611	6612	6615	6616
6617	6623	6624	6630	6631	6637	6638	6644	6645	6665	6669	6674	6675	6678	6680
6683	6684	6685	6709	6711	6713	6717	6718	6719	6720	6722	6724	6726	6729	6730
6732	6733	6736	6739	6742	6754	6755	6768	6770	6783	6784	6797	6799	6804	6805
6806	6811	6815	6817	6819	6829	6830	6838	6846	6847	6855	6865	6866	6874	6882
6883	6891	6901	6902	6910	6918	6919	6927	6937	6938	6946	6954	6955	6963	6966
6968	6972	6980	6982	6995	7003	7005	7018	7026	7028	7041	7049	7051	7074	7077
7078	7080	7081	7087	7091	7092	7095	7096	7166	7171	7172	7173	7174	7175	7178
7179	7180	7181	7182	7183	7184	7186	7187	7194	7196	7197	7198	7199	7200	7209
7211	7214	7223	7233	7234	7235	7302	7303	7304	7311	7316	7317	7319	7320	7323
7329	7330	7331	7332	7333	7334	7335	7336	7337	7338	7340	7341	7342	7343	7344
7345	7346	7347	7348	7349	7450	7451	7452	7470	7471	7472	7479	7484	7485	7487
7488	7491	7497	7498	7499	7500	7501	7502	7503	7504	7505	7506	7508	7509	7510
7511	7512	7513	7514	7515	7516	7517	7618	7619	7620	7638	7639	7640	7647	7652
7653	7655	7656	7659	7665	7666	7667	7668	7669	7670	7672	7673	7674	7675	7676
7677	7678	7679	7680	7681	7782	7783	7784	7803	7804	7805	7812	7817	7818	7820
7821	7824	7830	7831	7832	7833	7834	7835	7836	7837	7838	7839	7841	7842	7843
7844	7845	7846	7847	7848	7849	7850	7951	7952	7953	7969	7970	7971	7977	7980
7981	7982	7983	7984	7985	7988	7989	7991	7992	7995	7996	7997	8051	8052	8078
8081	8082	8083	8084	8085	8086	8087	8088	8091	8092	8094	8095	8099	8113	8114
8123	8124	8126	8128	8129	8130	8143	8147	8148	8150	8151	8152	8156	8157	8164
8165	8187	8199	8200	8206	8207	8213	8214	8233	8234	8253	8254	8255	8276	8277
8278	8299	8300	8301	8324	8325	8326	8328	8330	8334	8336	8337	8338	8341	8343
8356	8357	8359	8376	8377	8379	8394	8395	8397	8429	8433	8436	8438	8440	8475
8476	8478	8481	8486	8496	8508	8509	8511	8512	8515	8516	8537	8542	8555	8558
8564	8586	8587	8588	8589	8590	8591	8592	8593	8594	8595	8602	8603	8604	8605
8606	8607	8608	8609	8610	8611	8635	8636	8640	8655	8710	8718	8719	8720	8726
8733	8751	8752	8753	8754	8755	8776	8777	8778	8779	8780	8781	8782	8787	8790
8810	8816	8817	8818	8819	8820	8822	8823	8843	8844	8845	8846	8847	8848	8859
8860	8861	8862	8863	8876	8677	8881	8922	8923	8924	8925	8926	8927	8928	8929
8930	8931	8932	8938	8939	8943	8944	8945	8946	8947	8948	8949	8950	8951	8954
8973	8974	8975	8976	8977	8978	8981	8986	9004	9005	9006	9012	9013	9016	9019
9020	9021	9022	9023	9034	9048	9049	9050	9051	9056	9134	9151	9159	9160	9161
9162	9164	9172	9208	9209	9210	9211	9213	9222	9224	9225	9234	9238	9245	9251
9257	9258	9261	9264	9269	9275	9278	9285	9292	9293	9298	9311	9313	9328	9329
9347	9348	9365	9372	9373	9374	9377	9381	9415	9416	9427	9438	9442	9445	9456
9466	9476	9501	9504	9505	9506	9507	9520	9521	9548	9549	9575	9576	9610	9701
9702	9705	9708	9709	9710	9711	9712	9713	9724	9725	9726	9727	9728	9729	9730
9731	9732	9733	9734	9750	9752	9757	9758	9759	9760	9762	9768	9774	9777	9785
9787	9788	9791	9792	9793	9799	9814	9819	9820	9821	9822	9827	9859	9860	9863
9866	9867	9868	9869	9870	9871	9882	9883	9884	9885	9886	9887	9888	9889	9890
9891	9908	9910	9915	9916	9917	9918	9920	9926	9932	9937	9938	9943	9949	9951
9952	9955	9956	9957	9963	9976	9977	9978	9979	9984	10018	10019	10022	10025	10026
10027	10028	10029	10030	10047	10048	10049	10052	10053	10054	10055	10056	10072	10077	10078
10081	10082	10083	10085	10092	10095	10098	10101	10106	10109	10110	10111	10119	10124	10125

F07


```
        10127  10128  10129  10130  10134  10139  10144  10146  10151  10163  10164  10165  10171  10204  10205
        10208  10210  10211  10212  10213  10214  10215  10216  10224  10225  10227  10228  10229  10230  10231
        10232  10233  10248  10249  10253  10254  10255  10257  10262  10265  10266  10268  10269  10271  10272
        10285  10292  10293  10294  10333  10334  10337  10340  10341  10342  10343  10344  10345  10353  10354
        10355  10356  10357  10358  10360  10361  10362  10363  10364  10382  10384  10389  10390  10391  10392
        10394  10401  10407  10411  10413  10414  10417  10418  10420  10426  10440  10441  10442  10443  10448
MOVB     2168   2288   2513   2726   2926   3111   3183   3314   3596   3700   3833   4100   4394   4674   4685
         4696   4708   4720   4807   4945   4980   4992   5004   5016   5102   5277   5343   5527   5643   5713
         5820   5974   6119   6372   6714   6807   6812   6826   6843   6862   6879   6898   6915   6934   6951
         7169   7207   7324   7325   7326   7327   7492   7493   7494   7495   7660   7661   7662   7663   7825
         7826   7827   7828   8190   8192   8193   8194   8352   8364   8374   8383   8392   8401   8514   8544
         8637   8665   8673   8711   8712   8715   8716   8717   8721   8724   8725   8744   8785   8788   8802
         8805   8814   8879   8980   9035   9236   9542   9570   9597   9605   9706   9707   9735   9864   9865
         9892  10023  10024  10338  10339  10365
NEG      8722   8784   9781   9946  10122  10408
NOP      2415   2645   2836   3033   3482   3684   3963   4684   4695   4707   4719   4818   4830   4842   4854
         4979   4991   5003   5015   5139   5151   5163   5175   5294   5381   5383   5399   5415   5431   5446
         5547   5565   5573   5595   5690   5908   6062   6246   6247   6258   6259   6266   6267   6278   6279
         6286   6287   6298   6299   6410   6738   7240   7244   7301   7306   7307   7360   7372   7378   7392
         7397   7412   7429   7434   7441   7444   7469   7474   7475   7528   7540   7546   7560   7565   7580
         7597   7602   7609   7612   7637   7642   7643   7692   7704   7710   7724   7729   7744   7761   7766
         7773   7776   7802   7807   7808   7861   7873   7879   7893   7898   7913   7930   7935   7942   7945
         8349   8350   8355   8360   8373   8391   8444   8445   8446  10359
RESET    8442   9026
ROL      2466   2470   2555   2685   2888   2892   2960   3074   6808   6813   8728   8730   8731   8732   8734
         8850   9216
ROR      8990   8991   8992   8993   8994   8995   9360   9361   9362
RTI      2182   5319   7245   7375   7395   7415   7432   7439   7456   7543   7563   7583   7600   7607   7624
         7707   7727   7747   7764   7771   7788   7876   7896   7916   7933   7940   7957   8020   8036   8043
         8056   8106   8517   8566   8596   8612   8642   8756   8824   8956   9078   9085   9092   9099   9106
         9194   9226   9739   9743   9825   9896   9901   9982  10063  10068  10075  10168  10237  10242  10297
        10369  10374  10446
RTS      3574   4787   4922   5083   5243   6690   7459   7627   7791   7960   8057   8131   8168   8251   8274
         8297   8345   8682   8864   8882   8988   9141   9167   9330   9383   9478   9511   9543   9571   9598
         9606   9611   9763   9769   9772   9779   9783   9815   9830   9921   9927   9930   9944   9948   9972
         9997  10086  10093  10099  10120  10136  10160  10174  10258  10263  10281  10289  10302  10395  10402
        10405  10409  10435  10451
SBC      2396   2626   2818   3015   3919
SOB      2214   2300   2527   2737   2935   3204   3328   3563   3572   3615   3635   3842   4120   4132   4416
         4428   4964   5124   5302   5540   5557   5568   5591   5615   5658   5675   5709   6150   6165   6809
         6814   7176   7219   7241   8102   8117   8162   8228   8332   8340   8368   8387   8405   9163   9217
         9509   9540   9568   9595  10133  10275
SPL      7282   7315   7321   7361   7381   7401   7418   7448   7454   7483   7489   7529   7549   7569   7586
         7616   7622   7651   7657   7693   7713   7733   7750   7780   7786   7816   7822   7862   7882   7902
         7919   7949   7955   7987   7993   8008   8024   8049   8055   8077   8104   8142   8167  10073
SUB      2395   2625   2817   3014   3918   8127   8543   8791
SWAB     9939
TRAP     8884   8893   8894   8895   8896   8899   8900   8902   8903   8904   8905   8906   8907   8908   8909
         8910   8911   8913   8914   8915   8916   8917
TST      2443   2452   2459   2461   2671   2679   2866   2874   2880   2882   3060   3068   3121   3192   3361
         3391   3393   3462   3463   3483   3509   3571   3612   3614   3685   3686   3698   3704   3755   3781
         3938   3939   3980   3995   4117   4119   4197   4198   4269   4285   4316   4331   4350   4355   4413
         4415   4494   4495   4567   4583   4612   4627   4645   4649   4690   4701   4713   4725   4743   4878
         4985   4997   5009   5021   5039   5199   5301   5304   5373   5378   5537   5539   5549   5550   5556
         5567   5614   5654   5656   5667   5668   5704   5711   5742   5774   5833   5834   5879   5890   5987
         5988   6033   6034   6146   6148   6157   6158   6250   6252   6254   6256   6270   6272   6274   6276
```

```
        6290    6292    6294    6296    6334    6412    6413    6415    6416    6420    6434    6449    6463    6477    6597
        6740    6741    6743    6744    6748    6762    6777    6791    6974    6997    7020    7043    7238    7283    7285
        7287    7289    7291    7445    7455    7613    7623    7777    7787    7946    7956    8046    8054    8209    8249
        8256    8272    8279    8295    8302    8329    8331    8339    8477    8503    8550    8556    8639    8647    8669
        8739    8796    8806    8878    9018    9074    9076    9083    9090    9097    9104    9324    9350    9418    9429
        9439    9457    9467    9535    9563    9590    9697    9753    9794    9818    9855    9511    9958    9975   10013
       10079   10154   10162   10199   10250   10291   10328   10385   10421   10439
TSTB    7204    8490    8631    8671    8798    8812    9262    9764    9766    9789    9796    9811    9828    9922    9924
        9953    9960    9970    9985   10068   10090   10140   10148   10156   10172   10260   10287   10300   10397   10399
       10415   10423   10433   10449
.ASCII  1699    1700   10473   10483   10503   10516   10542   10549   10552   10562   10574   10588   10601   10613   10627
       10663   10718   10803   10812   10826   10840   10874   10886   10907   10923   10979   10987   10992   11005   11027
       11036   11047   11055   11067   11089   11102   11116   11143   11151   11163   11171   11185   11193   11208   11224
       11232   11235   11247   11255   11259   11271   11287   11311   11320   11331   11339   11356   11364   11383   11391
       11398   11403   11411   11425   11433   11440   11445   11453   11467   11475   11482   11487   11496   11510   11518
       11525   11530   11538   11552   11560   11567   11572   11581   11594   11602   11609   11614   11622   11634   11642
       11649   11654   11663   11676   11684   11691   11696   11705   11786   11824   11836   11845   11851   11862   11873
       11902   11925   11945   11965   11987   11997   12074   12082   12091
.ASCIZ  1698    1701    2196    8451   10454   10456   10463   10471   10476   10491   10493   10497   10500   10506   10522
       10525   10530   10534   10537   10559   10569   10584   10597   10610   10623   10636   10640   10643   10650   10656
       10668   10676   10682   10688   10693   10699   10705   10711   10726   10730   10735   10741   10751   10757   10762
       10768   10774   10779   10785   10793   10816   10834   10848   10854   10864   10882   10894   10897   10915   10929
       10933   10940   10948   10957   10969   10997   11014   11044   11062   11076   11097   11110   11124   11133   11158
       11176   11198   11215   11242   11266   11279   11295   11327   11347   11373   11418   11460   11503   11545   11589
       11629   11671   11713   11717   11727   11740   11746   11752   11761   11767   11777   11794   11803   11815   11831
       11841   11854   11857   11867   11879   11885   11894   11908   11917   11933   11952   11957   11970   11977   12003
       12011   12026   12033   12043   12064   12077   12085   12096   12102   12109   12116   12124   12129
.BLKB   8999
.BLKW   3536    7069    8829
.BYTE   1626    1627    1632    1637    1641    1642    1650    1651    1652    1653    7363    7364    7383    7384    7403
        7404    7421    7422    7531    7532    7551    7552    7571    7572    7589    7590    7695    7696    7715    7716
        7735    7736    7753    7754    7864    7865    7884    7885    7904    7905    7922    7923    8240    8241    8263
        8264    8286    8287    8316    8317    8318    8319    8320    8450    8757    8758    8759    8760    9287    9288
        9480    9481    9482    9483    9484    9526    9527    9554    9555    9581    9582   12137   12140   12147   12157
       12161   12164   12167   12170   12172   12175   12186   12188   12191   12195   12197   12204   12207   12218   12223
       12226   12230   12235   12241   12249   12264   12266   12271   12288   12291   12296
.ENABL     1    1159
.END   12512
.ENDC   1168    1186    1188    1189    1190    1577    1605    1609    1613    1615    1620    1624    1626    1654    1676
        1696    1697    1698    1699    1703    1856    1859    1862    1865    1868    1871    1874    1877    1880    1883
        1886    1889    1892    1895    1898    1901    1905    1908    1911    1914    1917    1920    1923    1926    1929
        1932    1935    1938    1941    1944    1948    1951    1954    1957    1960    1963    1966    1969    1972    1975
        1978    1981    1984    1987    1991    1994    1997    2000    2003    2006    2009    2012    2015    2018    2021
        2024    2027    2030    2033    2036    2039    2042    2045    2048    2051    2054    2057    2060    2063    2155
        2156    2159    2161    2163    2165    2166    2167    2169    2171    2187    2191    2193    2194    2196    2225
        2226    2281    2282    2283    2284    2288    2334    2335    2336    2353    2354    2355    2356    2364    2365
        2366    2431    2432    2433    2486    2487    2506    2507    2508    2509    2513    2574    2575    2576    2593
        2594    2595    2659    2660    2661    2708    2709    2719    2720    2721    2722    2726    2770    2771    2772
        2789    2790    2791    2792    2800    2801    2802    2853    2854    2855    2908    2909    2919    2920    2921
        2922    2926    2978    2979    2980    2996    2997    2998    3047    3048    3049    3097    3098    3104    3105
        3106    3107    3111    3170    3171    3176    3177    3178    3179    3183    3288    3289    3308    3309    3310
        3311    3314    3407    3408    3413    3435    3436    3441    3555    3556    3557    3563    3580    3581    3588
        3589    3590    3591    3596    3661    3662    3663    3736    3737    3738    3791    3792    3826    3827    3828
        3829    3833    3885    3886    3887    3888    3898    3899    3900    4042    4043    4044    4069    4070    4091
        4092    4093    4094    4099    4159    4160    4161    4162    4170    4171    4172    4231    4232    4233    4234
        4242    4243    4244    4363    4364    4385    4386    4387    4388    4393    4456    4457    4458    4459    4467
```

```
        4468  4469  4529  4530  4531  4532  4540  4541  4542  4657  4658  4667  4668  4669  4670
        4674  4794  4795  4800  4801  4802  4803  4807  4928  4929  4938  4939  4940  4941  4945
        5089  5090  5095  5096  5097  5098  5102  5250  5251  5271  5272  5273  5277  5331  5332
        5336  5337  5338  5339  5342  5515  5516  5520  5521  5522  5523  5526  5631  5632  5636
        5637  5638  5639  5642  5798  5799  5813  5814  5815  5816  5820  5823  5841  5845  5887
        5891  5952  5953  5967  5968  5969  5970  5974  5977  5995  5999  6041  6045  6107  6108
        6112  6113  6114  6117  6345  6346  6365  6366  6367  6368  6371  6421  6427  6430  6435
        6442  6445  6450  6456  6459  6464  6471  6474  6530  6533  6550  6553  6569  6572  6589
        6592  6695  6696  6707  6708  6709  6710  6713  6749  6755  6758  6763  6770  6773  6778
        6784  6787  6792  6799  6802  6975  6982  6985  6998  7005  7008  7021  7028  7031  7044
        7051  7054  7107  7108  7163  7164  7165  7168  7339  7507  7671  7840  8177  8178  8184
        8185  8186  8189  8414  8416  8417  8419  8422  8428  8431  8432  8436  8438  8440  8442
        8448  8450  8451  8454  8458  8461  8466  8470  8472  8483  8486  8489  8490  8492  8494
        8501  8505  8510  8511  8515  8518  8519  8523  8526  8535  8542  8547  8548  8549  8550
        8560  8564  8567  8571  8617  8637  8688  8766  8834  8871  8877  8880  8892  8893  8894
        8895  8896  8897  8898  8899  8900  8901  8902  8903  8904  8905  8906  8907  8908  8909
        8910  8911  8913  8914  8915  8916  8917  8921  8930  8931  8937  8943  8944  8954  8956
        8960  8964  9615  9673
.EQUIV  1199  1200  1202  1224  1225  1226  1227  1228  1229  1230  1231  1232  1233  1234  1263
        1264  1265  1266  1267  1268  1269  1270  1271  1272  1291  1292  1293  1294  1295  1296
        1297  1298  1299  1300  1350  1351  1352  1353  1560  1561  1562  1563  1564  1565  1566
        1567  1568  1569  1570  1571  1572  1573  1574  1575
.EVEN   2196  8321  9485 12306
.IF     1164  1185  1187  1188  1189  1190  1577  1603  1608  1611  1613  1619  1623  1625  1654
        1676  1696  1697  1698  1702  1703  1853  1856  1859  1862  1865  1868  1871  1874  1877
        1880  1883  1886  1889  1892  1895  1898  1903  1906  1909  1912  1915  1918  1921  1924
        1927  1930  1933  1936  1939  1942  1946  1949  1952  1955  1958  1961  1964  1967  1970
        1973  1976  1979  1982  1985  1989  1992  1995  1998  2001  2004  2007  2010  2013  2016
        2019  2022  2025  2028  2031  2034  2037  2040  2043  2046  2049  2052  2055  2058  2061
        2150  2155  2157  2159  2161  2163  2165  2166  2167  2169  2187  2190  2191  2192  2194
        2195  2224  2226  2281  2283  2284  2286  2330  2335  2336  2348  2354  2355  2356  2360
        2365  2366  2427  2432  2433  2485  2487  2506  2508  2509  2511  2570  2575  2576  2589
        2594  2595  2655  2660  2661  2707  2709  2719  2721  2722  2724  2766  2771  2772  2784
        2790  2791  2792  2796  2801  2802  2849  2854  2855  2907  2909  2919  2921  2922  2924
        2974  2979  2980  2992  2997  2998  3043  3048  3049  3096  3098  3104  3106  3107  3109
        3169  3171  3176  3178  3179  3181  3287  3289  3308  3310  3311  3312  3407  3408  3409
        3435  3436  3437  3555  3556  3557  3558  3579  3581  3588  3590  3591  3594  3657  3662
        3663  3732  3737  3738  3790  3792  3826  3828  3829  3831  3880  3886  3887  3888  3894
        3899  3900  4038  4043  4044  4068  4070  4091  4093  4094  4097  4154  4160  4161  4162
        4166  4171  4172  4226  4232  4233  4234  4238  4243  4244  4362  4364  4385  4387  4388
        4391  4451  4457  4458  4459  4463  4468  4469  4524  4530  4531  4532  4536  4541  4542
        4656  4658  4667  4669  4670  4672  4793  4795  4800  4802  4803  4805  4927  4929  4938
        4940  4941  4943  5088  5090  5095  5097  5098  5100  5249  5251  5271  5273  5275  5330
        5332  5336  5338  5339  5340  5514  5516  5520  5522  5523  5524  5630  5632  5636  5638
        5639  5640  5797  5799  5813  5815  5816  5818  5823  5841  5844  5887  5890  5951  5953
        5967  5969  5970  5972  5977  5995  5998  6041  6044  6106  6108  6112  6114  6115  6344
        6346  6365  6367  6368  6369  6420  6427  6430  6434  6441  6445  6449  6456  6459  6463
        6470  6474  6530  6533  6549  6553  6569  6572  6588  6592  6694  6696  6707  6709  6710
        6711  6748  6755  6758  6762  6769  6773  6777  6784  6787  6791  6798  6802  6974  6981
        6985  6997  7004  7008  7020  7027  7031  7043  7050  7054  7106  7108  7163  7165  7166
        7335  7503  7671  7836  8176  8178  8184  8186  8187  8413  8414  8415  8416  8417  8418
        8419  8421  8427  8430  8432  8436  8438  8440  8448  8450  8451  8457  8460  8465  8470
        8482  8484  8485  8486  8490  8491  8492  8501  8503  8511  8512  8517  8518  8519  8522
        8525  8535  8538  8545  8547  8548  8550  8553  8560  8564  8567  8570  8616  8637  8687
        8765  8833  8870  8876  8880  8884  8893  8894  8895  8896  8897  8898  8899  8900  8901
        8902  8903  8904  8905  8906  8907  8908  8909  8910  8911  8913  8914  8915  8916  8917
```

```
        8920  8930  8931  8936  8943  8944  8952  8954  8956  8960  8963  9614  9672
.IFF    1185  1188  1189  1190  1609  1613  1615  1620  1624  1625  1654  1703  1856  1859  1862
        1865  1868  1871  1874  1877  1880  1883  1886  1889  1892  1895  1898  1901  1905  1908
        1911  1914  1917  1920  1923  1926  1929  1932  1935  1938  1941  1944  1948  1951  1954
        1957  1960  1963  1966  1969  1972  1975  1978  1981  1984  1987  1991  1994  1997  2000
        2003  2004  2007  2010  2013  2016  2019  2022  2025  2028  2031  2034  2037  2040  2043
        2046  2049  2052  2057  2058  2061  2155  2190  2192  2225  2226  2292  2263  2288  2486
        2487  2507  2508  2513  2708  2709  2720  2721  2726  2908  2909  2920  2921  2926  3097
        3098  3105  3106  3111  3170  3171  3177  3178  3183  3288  3289  3309  3310  3314  3580
        3581  3589  3590  3596  3791  3792  3827  3828  3833  4069  4070  4092  4093  4099  4363
        4364  4386  4387  4393  4657  4658  4668  4669  4674  4794  4795  4801  4802  4807  4928
        4929  4939  4940  4945  5089  5090  5096  5097  5102  5250  5251  5272  5273  5277  5331
        5332  5337  5338  5342  5515  5516  5521  5522  5526  5631  5632  5637  5638  5642  5798
        5799  5814  5815  5820  5952  5953  5968  5969  5974  6107  6108  6113  6114  6117  6345
        6346  6366  6367  6371  6420  6435  6449  6464  6695  6696  6708  6709  6713  6748  6763
        6777  6792  6975  6998  7021  7044  7107  7108  7164  7165  7168  8177  8178  8185  8166
        8187  8414  8418  8422  8427  8430  8450  8458  8483  8486  8492  8518  8523  8525  8538
        8560  8564  8571  8617  8688  8766  8834  8871  8877  8921  8937  8952  8964  9615  9673
.IFT    2196  8500  8548
.IFTF   2196  8498  8547
.IIF    1163  1168  1173  1174  1182  1183  1184  1186  1189  1190  1191  1600  1702  2156  2159
        2165  2166  2167  2169  2170  2191  8416  8422  8423  8434  8450  8454  8461  8462  8463
        8464  8465  8466  8470  8499  8500  8515  8518  8519  8526  8527  8528  8529  8530  8535
        8553  8560  8567  8684  8892  8893  8894  8895  8896  8899  8900  8902  8903  8904  8905
        8906  8907  8908  8909  8910  8911  8913  8914  8915  8916  8917
.IRP    1577  1853  1902  1945  1988  2036  2224  2485  2707  2907  3096  3169  3287  3579  3790
        4068  4362  4656  4793  4927  5088  5249  5330  5514  5630  5797  5951  6106  6344  6694
        7106  8176  8438  8564  8586  8606  8776  8816  8843  8861  8924  8930  8943  8944
.LIST      1  1114  1189  1576  1577  1600  1654  1656  1657  1658  1659  1660  1661  1662  1663
        1664  1665  1666  1667  1668  1669  1670  1671  1672  1673  1674  1675  1676  1677  1678
        1679  1680  1681  1682  1683  1684  1685  1686  1687  1688  1689  1690  1691  1692  1693
        1694  1695  1696  2171  2191  2194  2196  2224  2283  2485  2508  2707  2721  2907  2921
        3096  3106  3169  3178  3287  3310  3579  3590  3790  3828  4068  4093  4362  4387  4656
        4669  4793  4802  4927  4940  5088  5097  5249  5273  5330  5338  5514  5522  5630  5638
        5797  5815  5951  5969  6106  6114  6344  6367  6694  6709  7106  7165  8176  8186  8422
        8438  8440  8465  8560  8884  8892  8893  8894  8895  8896  8897  8899  8900  8901  8902
        8903  8904  8905  8906  8907  8908  8909  8910  8911  8912  8913  8914  8915  8916  8917
        8918
.MACRO     1  1116  1117  1119  1120  1121  1122  1124  1126  1127  1128  1129  1130  1131  1132
        1133  1135  1136  1137  1138  1139  1140  1141  1142  1143  1144  1145  1146  1147  1148
        1149  1150  1151  1152  1153  1154  1155  1156  1157  1190  1617  2224  2484  2706  3096
        3168  3286  3578  3790  4067  4361  4655  4792  5248  5330  5514  5630  5796  5797  5951
        6105  6344  6693  7105  8175  8884 11382 11551
.MCALL  1160  1161  1162  1576  2171  2194
.NLIST     1  1115  1189  1576  1577  1600  1654  1656  1657  1658  1659  1660  1661  1662  1663
        1664  1665  1666  1667  1668  1669  1670  1671  1672  1673  1674  1675  1676  1677  1678
        1679  1680  1681  1682  1683  1684  1685  1686  1687  1688  1689  1690  1691  1692  1693
        1694  1695  1696  2171  2191  2194  2196  2224  2283  2485  2508  2707  2721  2907  2921
        3096  3106  3169  3178  3287  3310  3579  3590  3790  3828  4068  4093  4362  4387  4656
        4669  4793  4802  4927  4940  5088  5097  5249  5273  5330  5338  5514  5522  5630  5638
        5797  5815  5951  5969  6106  6114  6344  6367  6694  6709  7106  7165  8176  8186  8422
        8438  8440  8465  8560  8884  8892  8893  8894  8895  8896  8897  8899  8900  8901  8902
        8903  8904  8905  8906  8907  8908  8909  8910  8911  8912  8913  8914  8915  8916  8917
        8918
.PAGE   1617  1703
.REM       1
```

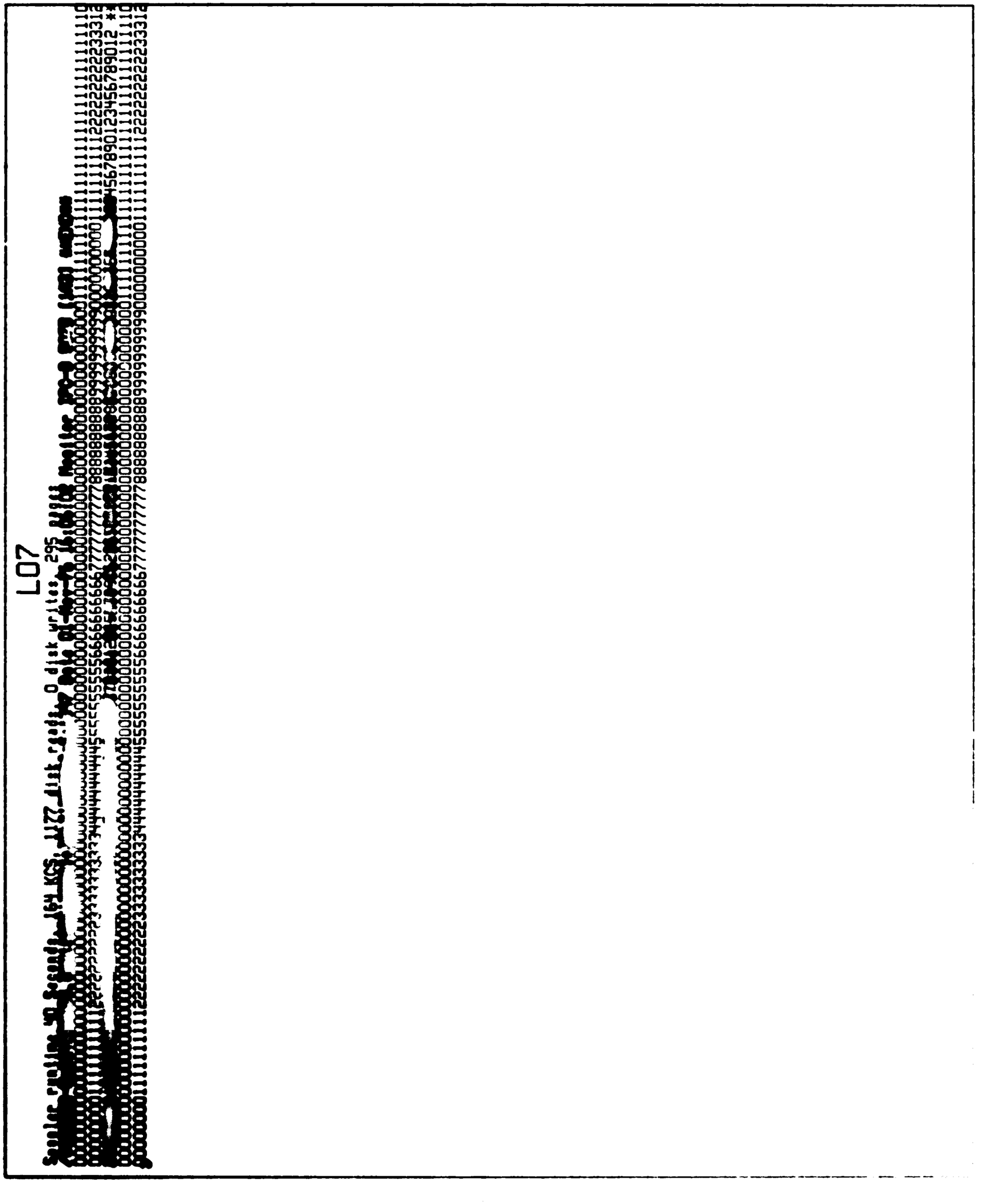
.REPT 1600 1656 1676
.SBTTL 1178 1192 1318 1328 1341 1489 1594 1603 1606 1617 1703 2149 2187 2224 2485
2707 2907 3096 3169 3287 3579 3790 4068 4362 4656 4793 4927 5088 5249 5330
5514 5630 5797 5951 6106 6344 6694 7106 8176 8411 8455 8520 8568 8614 8685
8763 8831 8868 8884 8918 8961 9385 9613 9674 9832 9990 10176 10304
.TITLE 1163
.WORD 1600 1601 1602 1614 1625 1628 1629 1630 1631 1634 1635 1636 1637 1638 1639
1640 1643 1644 1645 1654 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665
1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680
1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695
1722 1725 1728 1731 1734 1737 1740 1743 1746 1749 1752 1755 1758 1761 1764
1767 1770 1773 1776 1779 1782 1785 1788 1791 1794 1797 1800 1803 1806 1809
1812 1815 1818 1821 1824 1827 1830 1833 1836 1839 1842 1845 1848 1851 1854
1857 1860 1863 1866 1869 1872 1875 1878 1881 1884 1887 1890 1893 1896 1899
1903 1906 1909 1912 1915 1918 1921 1924 1927 1930 1933 1936 1939 1942 1946
1949 1952 1955 1958 1961 1964 1967 1970 1973 1976 1979 1982 1985 1989 1992
1995 1998 2001 2004 2007 2010 2013 2016 2019 2022 2025 2028 2031 2034 2037
2040 2043 2046 2049 2052 2055 2058 2061 2063 2064 2065 2066 2067 2068 2069
2070 2071 2072 2073 2074 2075 2076 2077 2079 2080 2081 2082 2083 2084 2085
2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100
2101 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2117
2118 2119 2121 2122 2123 2124 2125 2126 2127 2128 2131 2132 2133 2134 2135
2136 2137 2140 2141 2142 2143 2144 2315 2316 2475 2476 2477 2478 2479 2480
2481 2482 2539 2540 2695 2696 2697 2698 2699 2700 2702 2703 2751 2752 2898
2999 2900 2901 2902 2903 2946 2947 3084 3085 3086 3087 3088 3089 3091 3092
3116 3117 3164 3188 3189 3282 3340 3341 3517 3519 3525 3526 3527 3528 3529
3531 3532 3537 3601 3602 3711 3712 3713 3714 3715 3716 3718 3720 3723 3724
3726 3727 3853 3854 4003 4006 4010 4012 4013 4014 4015 4016 4017 4018 4019
4020 4021 4022 4105 4106 4292 4295 4297 4299 4300 4301 4302 4303 4304 4306
4307 4308 4401 4402 4590 4591 4592 4594 4595 4596 4597 4598 4599 4601 4602
4604 4730 4731 4732 4865 4866 4867 5026 5027 5028 5186 5187 5188 5291 5307
5323 5325 5499 5501 5502 5503 5504 5505 5507 5508 5623 5625 5626 5726 5727
5729 5731 5732 5734 5736 5738 5922 6076 6226 6227 6230 6232 6233 6234 6235
6236 6237 6238 6649 6652 6655 6656 6657 6658 6659 6661 6662 7063 7064 7065
7066 7068 7249 7252 7254 7255 7256 7257 7258 7260 7261 7262 7263 7264 7267
7268 7269 7270 7271 7273 7274 7275 7276 7277 7310 7354 7355 7356 7357 7365
7366 7367 7368 7369 7370 7385 7386 7387 7388 7389 7390 7405 7406 7407 7408
7409 7410 7423 7424 7425 7426 7427 7428 7461 7462 7478 7522 7523 7524 7525
7533 7534 7535 7536 7537 7538 7553 7554 7555 7556 7557 7558 7573 7574 7575
7576 7577 7578 7591 7592 7593 7594 7595 7596 7629 7630 7646 7686 7687 7688
7689 7697 7698 7699 7700 7701 7702 7717 7718 7719 7720 7721 7722 7737 7738
7739 7740 7741 7742 7755 7756 7757 7758 7759 7760 7793 7794 7811 7855 7856
7857 7858 7866 7867 7868 7869 7870 7871 7886 7887 7888 7889 7890 7891 7906
7907 7908 7909 7910 7911 7924 7925 7926 7927 7928 7929 7962 7963 7975 8002
8003 8004 8005 8010 8011 8012 8013 8014 8015 8016 8026 8027 8028 8029 8030
8031 8032 8059 8060 8108 8109 8110 8119 8120 8133 8134 8135 8136 8137 8242
8243 8244 8245 8246 8247 8265 8266 8267 8268 8269 8270 8288 8289 8290 8291
8292 8293 8308 8310 8311 8312 8427 8430 8449 8681 8761 8865 8866 8953 8955
9041 9060 9080 9087 9094 9101 9108 9113 9114 9115 9116 9117 9118 9119 9120
9121 9122 9123 9124 9158 9171 9178 9180 9233 9248 9250 9254 9256 9280 9300
9315 9321 9332 9338 9339 9379 9420 9431 9441 9448 9459 9469 9487 9488 9489
9490 9491 9513 9528 9529 9530 9531 9532 9533 9556 9557 9558 9559 9560 9561
9583 9584 9585 9586 9587 9588 9678 9679 9680 9681 9682 9683 9684 9685 9686
9687 9688 9689 9690 9691 9692 9693 9694 9695 9836 9837 9838 9839 9840 9841
9842 9843 9844 9845 9846 9847 9848 9849 9850 9851 9852 9853 9994 9995 9996
9997 9998 9999 10000 10001 10002 10003 10004 10005 10006 10007 10008 10009 10010 10011

K07

10180	10181	10182	10183	10184	10185	10186	10187	10188	10189	10190	10191	10192	10193	10194
10195	10196	10197	10309	10310	10311	10312	10313	10314	10315	10316	10317	10318	10319	10320
10321	10322	10323	10324	10325	10326	12310	12313	12321	12331	12336	12340	12343	12346	12348
12352	12355	12358	12362	12365	12368	12372	12374	12377	12381	12384	12387	12390	12393	12396
12399	12402	12405	12408	12411	12416	12419	12423	12428	12434	12437	12443	12446	12451	12454
12460	12463	12468	12471	12476	12493	12496	12501	12510						

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DEKBDC.SEQ/SOL/CRF/PAGNUM/NL:TOC/DS:ERFZ=SYSMAC.CO,DEKBDC.P11
RUN-TIME: 76 115 25 SECONDS
RUN-TIME RATIO: 307/216=1.4
CORE USED: 44K (87 PAGES)


LO7