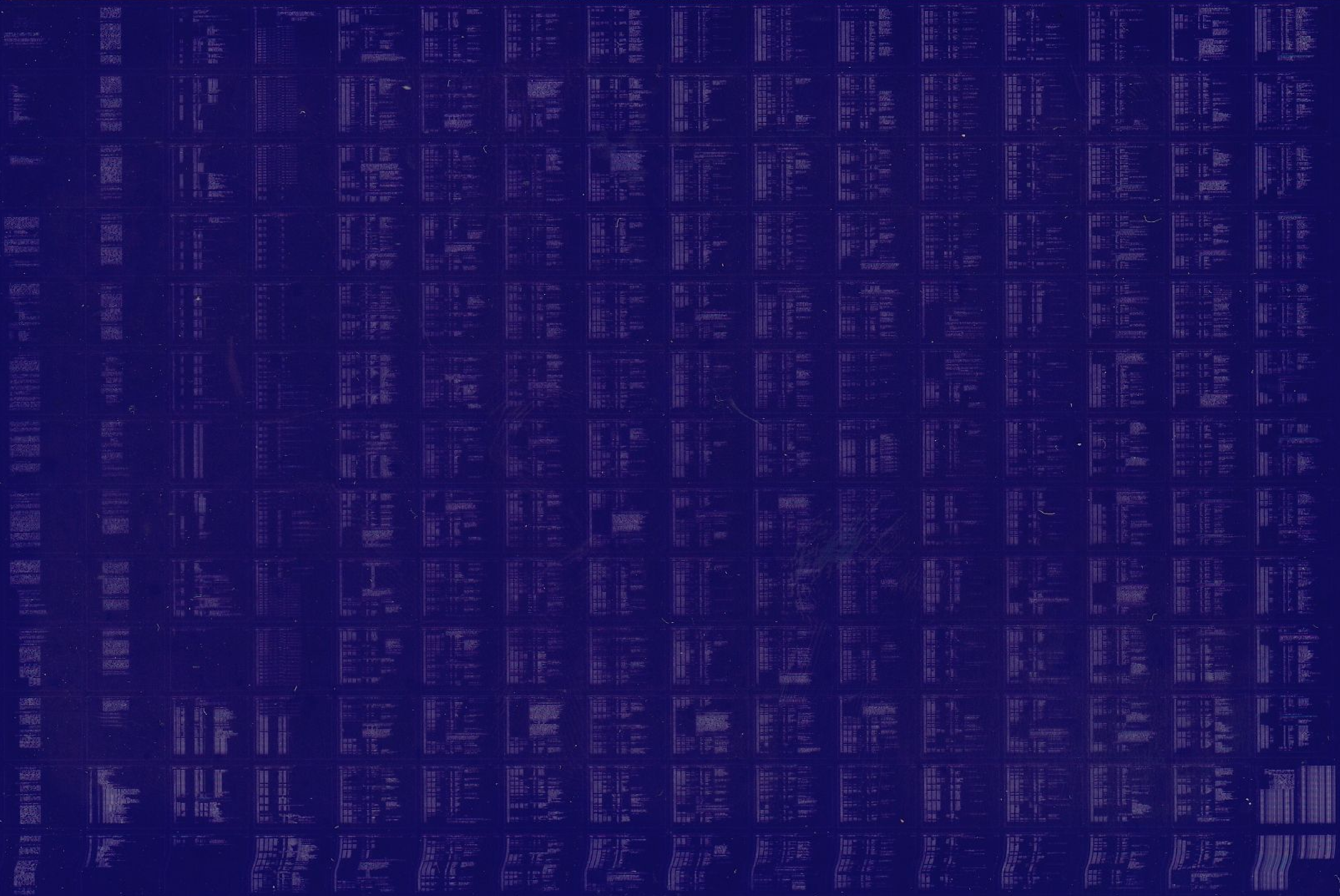
PDP11·70/74

11/70 CACHE #2
CEKBDEO

AH-7972E-MC
FICHE 1 OF 2

MAY 1980
COPYRIGHT© 75.80
MADE IN USA



PDP11·70/74 11/70 CACHE #2
CEKBDEO

AH·7972E·MC
FICHE 2 OF 2

MAY 1980
COPYRIGHT© 75·80
MADE IN USA

IDENTIFICATION

```
PRODUCT CODE:    AC-7971E-MC

PRODUCT NAME:    CEKBDEO 11/70 CACHE #2

DATE:            MAY, 1980

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

CONTENTS

```
        REVISION HISTORY
***************************************************************************
REV E0  1)PROGRAM MADE APT COMPATIBLE
        2)DIAGNOSTIC UTILIZATION OF MAP REGISTERS 0-2 RESTRICTED TO
          PASSIVE RELOCATION TO PREVENT APT/ACT INTERFERENCE
        3)TYPING CONTTROL-C WHILE IN AUTO MODE WILL RETURN CONTROL TO
          THE MONITOR RATHER THAN HALTING PROGRAM
        4)MEMORY SIZE ROUTINE WILL NOT ACCESS MORE THAN 1920K OF MEMORY
          TO PREVENT TEST 6 FAILURE ON SYSTEMS WITH >1920K
***************************************************************************
```

1. ABSTRACT

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

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

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

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

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

2. REQUIREMENTS

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

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

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

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

3. LOADING PROCEDURE

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

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS (SEE 5.1)

4.2 STARTING ADDRESS - 200

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

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

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

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS FOR CEKBC:

SW<15>=1	HALT ON ERROR
SW<14>=1	LOOP ON TEST
SW<13>=1	INHIBIT ERROR TYPOUTS
SW<12>	NOT USED IN CEKBC
SW<11>=1	INHIBIT ITERATIONS
SW<10>-1	RING BELL ON ERROR
SW<9> =1	LOOP ON ERROR
SW<8> -1	LOOP ON TEST IN SW<6:0>
SW<7> =1	SKIP EXECUTION OF TESTS WHICH USE MEMORY MANAGEMENT.
SW<6:0>	TEST NUMBER FOR LOOPING WHEN SW<8>

CEKBD USES THE SAME SWITCH SETTINGS AS CEKBC EXCEPT:

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

5.2 SUBROUTINE ABSTRACTS - BOTH CEKBC AND CEKBD USE THE FOLLOWING SUBROUTINES.

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

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

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

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

5.2.5 TRAP - A NUMBER OF SUBROUTINES ARE CALLED BY USING THE TRAP INSTRUCTION:

```
TYPE     TO TYPE OUT AN ASCIZ STRING
TYPEOC   TO TYPE OUT THE OCTAL FOR A 16-BIT BINARY
NUMBER ETC.
```

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

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

THE TERMINAL AND THAT MONITOR OR LOADER WILL AUTOMATICALLY BE RESTORED. AFTER THIS IS DONE THE

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

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

6. ERRORS

6.1 ERROR HALTS - ONLY TEST NUMBER 14 IN PROGRAM CEKBC, THE MAINTENANCE REGISTER COUNT PATTERN TEST, HALTS THE PROCESSOR IN THE SITUATION WHERE IT CAN'T CLEAR THE MAINTENANCE REGISTER. HERE PROCEDING WITH THE PROGRAM'S EXECUTION WOULD PROBABLY BE FATAL, SO A HALT IS EXECUTED! NO OTHER TEST IN EITHER PROGRAM SHOULD HALT UNDER ANY NORMAL ERROR DETECTION.

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

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS - NONE

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

8. MISCELLANEOUS

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

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

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

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

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

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

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

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

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

9. PROGRAM DESCRIPTION

9.1 CEKBD

PROGRAM BY ANTHONY S. VEZZA

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

TEST 1 PARITY ERROR ABORT

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

TEST 2 PARITY ERROR TRAP

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

TEST 3 MEM MGT AND PE TRAP PRIORITY ARBITRATION

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

TEST 4 UNIBUS PARITY ERROR

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

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

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

THIS TEST IS A TEST OF BOTH THE AMX, CPU INPUTS, AND THE CACHE ERROR ADDRESS REGISTER. A SET OF ADDRESSES IS GENERATED AND A MAIN MEMORY ADDRESS AND CONTROL LINE PARITY ERROR IS FORCED AT EACH, THEREBY LOCKING UP THE ADDRESS ON THE OUTPUT OF THE AMX IN THE ERROR ADDRESS REGISTER. THE MANNER IN WHICH THIS IS DONE IS AS FOLLOWS: FIRST THE ADDRESS IS GENERATED; THEN, IF IT IS A VALID ADDRESS (THAT IS, IF IT IS NOT BEYOND THE LIMITS OF MEMORY AS DISPLAYED IN THE SYSTEM SIZE REGISTER), THESE THREE INSTRUCTIONS ARE MOVED TO THAT AREA OF MEMORY:

```
        ONE:    MOV     R1,(R2)
        2$:     CLR     (R2)
        3$:     RTS     PC 2$ IS THE
```

ADDRESS BEING TESTED. THE INSTRUCTION AT ONE IS GIVEN CONTROL BY A 'JSR PC'. R1 IS MADE TO CONTAIN #2 AND R2 CONTAINES THE ADDRESS OF THE MAINTENANCE REGISTER, SO THAT AFTER THE 'MOV R1,(R2)' IS EXECUTED A PARITY ERROR SHOULD OCCUR ON THE MAIN MEMORY ADDRESS AND CONTROL LINES WHEN THE NEXT INSTRUCTION IS FETCHED. THE ADDRESSES USED ARE GENERATED FOLLOWING THIS PATTERN

```
                200000 200002 200004
                200010 200020 200040
                200100 200200 200400
                ETC.    TO:   240000
                300000 400000 400002
                400004  400010  ETC.
                TO:   500000  600000
                1000000      1000002
                1000004 ETC.
```

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

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

THIS IS ANOTHER TEST OF THE AMX WHICH IS CARRIED OUT USING THE SAME METHOD AS IN THE PREVIOUS TEST ALL THAT IS DIFFERENT IS THE SERIES OF TEST ADDRESSES WHICH IS USED. IN THE PREVIOUS TEST A ONE WAS FLOATED THROUGH A FIELD OF ZEROES TO PRODUCE THE TEST ADDRESSES, HERE A ZERO WIL BE FLOATED THROUGH A FIELD OF ONES TO PRODUCE THE ADDRESSES BASE ADDRESSES WHICH ARE USE ARE:

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

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

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

THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX. THIS TEST IS IDENTICAL TO TST1 IN EVERY THING IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM THE CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE AS UNIBUS REFERENCES.

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

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

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

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

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

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

TEST 13 CACHE ADDRESS MEMORY COMPARATOR TEST

THIS IS A TEST OF THE CACHE ADDRESS MEMORY ADDRESS COMPARATORS. THIS IS A CIRCUIT MADE UP OF SIX 74585 CHIPS, THREE FOR EACH GROUP. EACH CHIP COMPARES FOUR BITS OF THE ADDRESS ON THE ADDRESS MULTIPLEXER, AMX, OUTPUT LINES WITH THE RESPECTIVE FOUR BITS FROM THE CACHE ADDRESS MEMORY. TWELVE BITS OF THE ADDRESS ARE BROKEN DOWN THUS: BITS 10 THROUGH 13 FOR THE FIRST COMPARATOR; BITS 14 THROUGH 17 FOR THE NEXT; AND BITS 18 THROUGH 21 FOR THE LAST. THE METHOD CHOSEN FOR THIS TEST IS TO TAKE EACH POSSIBLE 4-BIT INPUT CONDITION FOR A COMPARATOR FROM THE ADDRESS MEMORY AND PUT EVERY POSSIBLE 4-BIT COMBINATION ON THE AMX SIDE OF THE COMPARATOR. FOR 4-BITS THERE ARE 16

(DEC) CONDITIONS. THUS FOR EVERY 4-BIT ADDRESS MEMORY INPUT TO THE COMPARATOR THERE ARE 16 AMX INPUT COMBINATIONS ONE OF WHICH WILL CAUSE A MATCH AND MAKE THE REFERENCE A HIT. THE OTHER 15 SHOULD OF COURSE BE MISSES.

TEST 14 CACHE ADDRESS MEMORY COUNT PATTERN TEST

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

TEST 15 CACHE ADDRESS MEMORY PARITY LOGIC TEST

THIS IS A TEST OF THE PARITY CHECKERS AND PARITY GENERATOR OF THE CACHE ADDRESS MEMORY. EVERY POSSIBLE ADDRESS TAG, BITS 21 THROUGH 10, WHICH CAN BE STORED IN THE CACHE ADDRESS MEMORY IS GENERATED, MADE A HIT AND THE MAINTENANCE REGISTER IS THEN USED TO FORCE A CACHE ADDRESS MEMORY PARITY ERROR AT EACH OF THE ADDRESSES GENERATED. NOTE THAT BITS 9 THROUGH 0 OF THE ADDRESSES
IS NOT OF CONCERN, SO THESE BITS WILL BE THE SAME FOR EACH ADDRESS; THIS IS BECAUSE ONLY BITS 21 THROUGH 10 ARE STORED IN THE ADDRESS MEMORY THEREFORE ONLY THESE BITS ARE PARITY CHECKED IN THE CACHE ADDRESS MEMORY PARITY CHECKERS. ALSO NOTE THAT THE RANGE OF THE ADDRESSES MUST BE LIMITED TO BETWEEN THE BOUNDS IMPOSED BY THE HIGHEST AVAILABLE MFMORY WORD AND THE LAST WORD OF MEMORY USED BY THIS PROGRAM. THE MANNER IN WHICH THE ERROR WILL BE FORCED WILL BE TO PUT THE INSTRUCTIONS:

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

AT THE PARTICULAR ADDRESS BEING TESTED, WHERE 'TSTADS' IS THE ADDRESS BEING TESTED. R4 CONTAINS A PATTERN TO BE LOADED IN THE MAINTENANCE REGISTER

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

TEST '6 CACHE ADDRESS MEMORY DUAL ADDRESS TEST, UPWARD

THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS MEMORY. AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS' ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP BEING TESTED. OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY ON THE SYSTEM. NOTE THAT HERE THE WORD 'TAG' REFERS TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21, WHICH ARE STORED IN THE CACHE ADDRESS MEMORY. HERE THE ADDRESS MEMORY IS WRITTEN IN THE UPWARD DIRECTION, THAT IS 'TAG' 1 IS WRITTEN FIRST, 'TAG' 2 SECOND ETC. THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED TO SEE IF IT IS A HIT. THUS MAKING SURE NO 'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER 'TAG'. NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD INVOLVE WRITTING THE 'TAGS' IN THE DOWNWARD DIRECTION AS WELL AS THE UPWARD DIRECTION. THE DOWNWARD WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND IN TST13.

TEST 17 CACHE ADDRESS MEMORY DUAL ADDRESS TEST, DOWNWARD

THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS MEMORY. AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS' ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP BEING TESTED. OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY ON THE SYSTEM. NOTE THAT HERE THE WORD 'TAG' REFERS TO THAT PART OF AN

ADDRESS, BITS 10 THROUGH 21, WHICH ARE STORED IN THE CACHE ADDRESS MEMORY. HERE THE ADDRESS MEMORY IS WRITTEN IN THE DOWNWARD DIRECTION, THAT IS 'TAG' 256 IS WRITTEN FIRST, 'TAG' 255 SECOND ETC. THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED TO SEE IF IT IS A HIT, THUS MAKING SURE NO 'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER 'TAG'. NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD INVOLVE WRITTING THE 'TAGS' IN THE UPWARD DIRECTION AS WELL AS THE DOWNWARD DIRECTION. THE UPWARD WRITING PART OF THIS DUAL ADDRESS TFST IS FOUND IN TST12.

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

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

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

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

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

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

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

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

TEST 24 CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST

THIS TEST IS EXECUTED OPTIONALLY, ON THE CONDITION THAT BIT 12 OF THE SWITCH REGISTER IS ON WHEN PROGRAM CONTROL REACHES THIS POINT. IF THIS SWITCH IS OFF THEN CONTROL IS PASSED TO THE NEXT TEST. THIS IS DONE BECAUSE THIS TEST REQUIRES OPERATOR INTERVENTION. THE USER IS ASKED TO GO THROUGH A POWER DOWN-POWER UP SEQUENCE. THEN A SIMPLE SCAN IS MADE OF MEMORY WHICH CAUSES ALL DATA AND ADDRESS MEMORY LOCATIONS IN THE CACHE TO BE PARITY CHECKED. IF THE POWER UP-CACHE INVLIDATER LOGIC WORKED NO PARITY ERRORS CAN OCCUR. BUT IF THIS INVALIDATER FAILED THERE IS AN EXTREMELY HIGH PROBABILITY FOR THE OCCURENCE OF A CACHE DATA OR CACHE ADDRESS PARITY ERROR. IN FACT IF THE INVALIDATER CIRCUIT IS COMPLETELY INOPERATIVE IT WILL BE VIRTUALLY IMPOSSIBLE TO RESTART THE PROGRAM. WHEREAS MINOR OR NO FAIL'JRES CAN AND WILL BE REPORTED. IF NO PARITY ERRORS ARE ENCOUNTERED THE USER WILL BE NOTIFIED SO THAT HE CAN KNOW IF A FATAL FAILURE HAS OCCURRED.

TEST 25 CACHE DATA MULTIPLEXER, CDMX, TEST

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

TEST 26 CACHE DATA MEMORY ADDRESS DRIVERS TEST

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

TEST 27 CACHE DATA MEMORY COUNT PATTERN TEST

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

TEST 30 CACHE DATA MEMORY PARITY CHECKERS LOW BYTE TEST

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

TEST 31 CACHE DATA MEMORY PARITY CHECKERS HIGH BYTE TEST

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

TEST 32 CACHE DATA MEMORY WORST CASE NOISE TEST

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

TEST 33 CACHE DATA MEMORY CHIP SELECTION LOGIC TEST

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

SET 0 GROUP 0 EVEN WORD
SET 1 GROUP 0 ODD WORD
SET 2 GROUP 1 EVEN WORD
SET 3 GROUP 1 ODD WORD A
DIFFERENT PATTERN, 000000 177777 125252 AND 052525, IS WRITTEN INTO EACH GROUP AND THEN READ BACK.
EVERY PERMUTATION OF THE
FOUR TEST PATTERNS IN THE FOUR SETS IS TRIED AND CHECKED. FOR EACH PERMUTATION OF THE TEST PATTERNS THIS ROUTINE FIRST WRITES 'UP' (SET 0 FIRST THEN 1,2 AND 3) THEN 'DOWN' (SET 3 FIRST THEN 2,1 AND 0).

TEST 34 CACHE DATA MEMORY BYTE ENABLE LOGIC TEST

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

TEST 35 CACHE ARBITRATION AND HIGH SPEED I/O TEST

THIS IS A TEST OF:

1. CACHE ARBITRATION
2. THE MASS BUS AND UNIBUS PORTS TO THE CACHE
3. HIGH SPEED I/O THROUGH THE CACHE

IT MAKES USE OF THE FOLLOWING DEVICES:

1. RS04
2. RP04
3. RK05
4. MASS BUSS TESTER
5. UNIBUS EXERCISER

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

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

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

ALL AVAILABLE DEVICES ARE STARTED DOING TRANSFERS AT THE SAME TIME TO DIFFERENT PARTS OF MEMORY. EACH DEVICE HAS A CONTROL ROUTINE WHICH DRIVES THAT DEVICE THROUGH THE CYCLE:

1. WRITE A RANDOM DATA PATTERN IN MEMORY
2. COPY THAT PATTERN ONTO THE DISK
3. WRITE CHECK THE DISK

4. READ THE PATTERN OFF THE DISK BACK INTO MEMORY
5. CHECK DATA

6. START OVER AT 1.

EACH DEVICE IS CAUSED TO GO 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 36 MASS BUS CACHE WRITE HIT CYCLE, INVALIDATION TEST

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

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

TEST 37 CHECK IVSS, VSIU BITS

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

TEST 40 CHECK VSIU BIT, WITH IVSS ALREADY SET

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

TEST 41 CHECK VCIP SETS WHEN CF IS SET

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

TEST 42 CHECK CACHE FLUSH & VALID STORE SWITCHING

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

TEST 43 CHECK IVSS INHIBITS SWITCHING OF VALID STORE IN USE

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

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

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

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

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

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

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

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

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

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

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

TEST 51 CHECK CACHE-BYPASS

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

TEST 52 CHECK CACHE IS BYPASSED ON ASRB OPERAND

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

TEST 53 CHECK CACHE VALID STORE PARITY CHECKER

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

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

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

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

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

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

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

```
                .TITLE  CEKBD-E  11/70 CACHE #2
                ;*COPYRIGHT (C) 1975, 1980
                ;*DIGITAL EQUIPMENT CORP.
                ;*MAYNARD, MASS. 01754
                ;*
                ;*
                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                ;*
        000001  $TN=1
        160000  $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
        167400  $SWR=167400
        000200  $SWRMK=200

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

                .SBTTL  BASIC DEFINITIONS

                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
        001100  STACK=  1100            ;;FIRST ADDRESS OF THE STACK
        001100  KERSTK= STACK           ;;KERNEL STACK
        000700  SUPSTK= STACK-200       ;;SUPERVISOR STACK
        000600  USESTK= STACK-300       ;;USER STACK
                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
        177776  PS=     177776          ;;PROCESSOR STATUS WORD
                .EQUIV  PS,PSW
        177774  STKLMT= 177774          ;;STACK LIMIT REGISTER
        177772  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
        177570  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
000006          SP=     %6              ;;STACK POINTER
                .EQUIV  SP,KSP          ;;KERNEL STACK POINTER
                .EQUIV  SP,SSP          ;;SUPERVISOR STACK POINTER
                .EQUIV  SP,USP          ;;USER STACK POINTER
000007          PC=     %7              ;;PROGRAM COUNTER

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

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

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

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

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


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

                .SBTTL  CPU REGISTER DEFINITIONS


177760          SIZELO = 177760         ;;MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
```

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



                                  .SBTTL  MEMORY MANAGEMENT DEFINITIONS


                                  ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

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

                                  ;*USER "I" PAGE DESCRIPTOR REGISTERS

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

                                  ;*USER "D" PAGE DESCRIPTOR REGISTORS

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

                                  ;*USER "I" PAGE ADDRESS REGISTERS

        177640                    UIPAR0= 177640
        177642                    UIPAR1= 177642
        177644                    UIPAR2= 177644
        177646                    UIPAR3= 177646
        177650                    UIPAR4= 177650
        177652                    UIPAR5= 177652
        177654                    UIPAR6= 177654
        177656                    UIPAR7= 177656

```

```
                                  ;*USER ''D'' PAGE ADDRESS REGISTERS

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

                                  ;*SUPERVISOR ''I'' PAGE DESCRIPTOR REGISTERS

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

                                  ;*SUPERVISOR ''D'' PAGE DESCRIPTOR REGISTERS

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

                                  ;*SUPERVISOR ''I'' PAGE ADDRESS REGISTERS

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

                                  ;*SUPERVISOR ''D'' PAGE ADDRESS REGISTERS

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

                                  ;*KERNEL ''I'' PAGE DESCRIPTOR REGISTERS

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

                                ;*KERNEL 'D'' PAGE DESCRIPTOR REGISTERS

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

                                ;*KERNEL ''I'' PAGE ADDRESS REGISTERS

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

                                ;*KERNEL 'D'' PAGE ADDRESS REGISTERS

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



                                .SBTTL  UNIBUS MAP REGISTER DEFINITIONS


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


        170200                  MAPL00 = 170200
        170202                  MAPH00 = 170202
        170204                  MAPL01 = 170204
        170206                  MAPH01 = 170206
```

```
 337          170210                  MAPL02 = 170210
 338          170212                  MAPH02 = 170212
 339          170214                  MAPL03 = 170214
 340          170216                  MAPH03 = 170216
 341          170220                  MAPL04 = 170220
 342          170222                  MAPH04 = 170222
 343          170224                  MAPL05 = 170224
 344          170226                  MAPH05 = 170226
 345          170230                  MAPL06 = 170230
 346          170232                  MAPH06 = 170232
 347          170234                  MAPL07 = 170234
 348          170236                  MAPH07 = 170236
 349          170240                  MAPL10 = 170240
 350          170242                  MAPH10 = 170242
 351          170244                  MAPL11 = 170244
 352          170246                  MAPH11 = 170246
 353          170250                  MAPL12 = 170250
 354          170252                  MAPH12 = 170252
 355          170254                  MAPL13 = 170254
 356          170256                  MAPH13 = 170256
 357          170260                  MAPL14 = 170260
 358          170262                  MAPH14 = 170262
 359          170264                  MAPL15 = 170264
 360          170266                  MAPH15 = 170266
 361          170270                  MAPL16 = 170270
 362          170272                  MAPH16 = 170272
 363          170274                  MAPL17 = 170274
 364          170276                  MAPH17 = 170276
 365          170300                  MAPL20 = 170300
 366          170302                  MAPH20 = 170302
 367          170304                  MAPL21 = 170304
 368          170306                  MAPH21 = 170306
 369          170310                  MAPL22 = 170310
 370          170312                  MAPH22 = 170312
 371          170314                  MAPL23 = 170314
 372          170316                  MAPH23 = 170316
 373          170320                  MAPL24 = 170320
 374          170320                  MAPH24 = 170320
 375          170324                  MAPL25 = 170324
 376          170326                  MAPH25 = 170326
 377          170330                  MAPL26 = 170330
 378          170332                  MAPH26 = 170332
 379          170334                  MAPL27 = 170334
 380          170336                  MAPH27 = 170336
 381          170340                  MAPL30 = 170340
 382          170342                  MAPH30 = 170342
 383          170344                  MAPL31 = 170344
 384          170346                  MAPH31 = 170346
 385          170350                  MAPL32 = 170350
 386          170352                  MAPH32 = 170352
 387          170354                  MAPL33 = 170354
 388          170356                  MAPH33 = 170356
 389          170360                  MAPL34 = 170360
 390          170362                  MAPH34 = 170362
 391          170364                  MAPL35 = 170364
 392          170366                  MAPH35   170366
```

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

                        ;DEFINITIONS

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

100000                  BYP=BIT15

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

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








000011          TAB 11
```

```
        000044                  S1M0=44
        000030                  S0M1=30
        000054                  S1M0M1=54
        000034                  S0M0M1=34
        000014                  M1M0=14
        000014                  M0M1=M1M0
        140000                  TESTR1=140000
        142000                  TESTR2=142000
        144000                  TESTR3=144000
        001400                  STACK=1400
                                .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  004146                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

                                .SBTTL  ACT11 HOOKS


                                ;;*************************************************************
                                ;HOOKS REQUIRED BY ACT11
        000204                          $SVPC=.                 ;SAVE PC
        000046                          .=46
000046  051464                          $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

        001400                  .-1400
                                .SBTTL  APT PARAMETER BLOCK


                                ;;*************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;*************************************************************
        001400                          .$X=.   .;SAVE CURRENT LOCATION
        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                          200     ;;FOR APT START UP
        000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001400                          $APTHDR ;;POINT TO APT HEADER BLOCK
        001400                          .=.$X   ;;RESET LOCATION COUNTER
                                ;;*************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001400                          $APTHD:
001400  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001402  001716                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001404  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
001406  000025                  $PASTM: .WORD   25      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001410  000000                  $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001412  000014                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
   505
   506
   507
   508
```

```
                                .SBTTL  COMMON TAGS

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

                001500                  .=1500
        001500                  $CMTAG:                         ;;START OF COMMON TAGS
        001500  000000                  .WORD   0
        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)
```

```
001624 000000          $REG21: .WORD  0               ;;CONTAINS (($REGAD)+42)
001626 000000          $REG22: .WORD  0               ;;CONTAINS (($REGAD)+44)
001630 000000          $REG23: .WORD  0               ;;CONTAINS (($REGAD)+46)
001632 000000          $TMP0:  .WORD  0               ;;USER DEFINED
001634 000000          $TMP1:  .WORD  0               ;;USER DEFINED
001636 000000          $TMP2:  .WORD  0               ;;USER DEFINED
001640 000000          $TMP3:  .WORD  0               ;;USER DEFINED
001642 000000          $TMP4:  .WORD  0               ;;USER DEFINED
001644 000000          $TMP5:  .WORD  0               ;;USER DEFINED
001646 000000          $TMP6:  .WORD  0               ;;USER DEFINED
001650 000000          $TMP7:  .WORD  0               ;;USER DEFINED
001652 000000          $TMP10: .WORD  0               ;;USER DEFINED
001654 000000          $TMP11: .WORD  0               ;;USER DEFINED
001656 000000          $TMP12: .WORD  0               ;;USER DEFINED
001660 000000          $TMP13: .WORD  0               ;;USER DEFINED
001662 000000          $TMP14: .WORD  0               ;;USER DEFINED
001664 000000          $TMP15: .WORD  0               ;;USER DEFINED
001666 000000          $TMP16: .WORD  0               ;;USER DEFINED
001670 000000          $TMP17: .WORD  0               ;;USER DEFINED
001672 000000          $TMP20: .WORD  0               ;;USER DEFINED
001674 000000          $TMP21: .WORD  0               ;;USER DEFINED
001676 000000          $TMP22: .WORD  0               ;;USER DEFINED
001700 000000          $TMP23: .WORD  0               ;;USER DEFINED
001702 000000          $TIMES: 0                      ;;MAX. NUMBER OF ITERATIONS
001704 000000          $ESCAPE:0                      ;;ESCAPE ON ERROR ADDRESS
001706 177607 000377   $BELL:  .ASCIZ <207><377><377> ;;CODE FOR BELL
001712    077          $QUES:  .ASCII /?/             ;;QUESTION MARK
001713    015          $CRLF:  .ASCII <15>            ;;CARRIAGE RETURN
001714 000012          $LF:    .ASCIZ <12>            ;;LINE FEED
                       ;;*******************************************************************
                       .SBTTL  APT MAILBOX-ETABLE

                       ;;*******************************************************************
                       .EVEN
001716                 $MAIL:                  ;;APT MAILBOX
001716 000000          $MSGTY: .WORD  AMSGTY   ;;MESSAGE TYPE CODE
001720 000000          $FATAL: .WORD  AFATAL   ;;FATAL ERROR NUMBER
001722 000000          $TESTN: .WORD  ATESTN   ;;TEST NUMBER
001724 000000          $PASS:  .WORD  APASS    ;;PASS COUNT
001726 000000          $DEVCT: .WORD  ADEVCT   ;;DEVICE COUNT
001730 000000          $UNIT:  .WORD  AUNIT    ;;I/O UNIT NUMBER
001732 000000          $MSGAD: .WORD  AMSGAD   ;;MESSAGE ADDRESS
001734 000000          $MSGLG: .WORD  AMSGLG   ;;MESSAGE LENGTH
001736                 $ETABLE:                ;;APT ENVIRONMENT TABLE
001736    000          $ENV:   .BYTE  AENV     ;;ENVIRONMENT BYTE
001737    000          $ENVM:  .BYTE  AENVM    ;;ENVIRONMENT MODE BITS
001740 000000          $SWREG: .WORD  ASWREG   ;;APT SWITCH REGISTER
001742 000000          $USWR:  .WORD  AUSWR    ;;USER SWITCHES
001744 000000          $CPUOP: .WORD  ACPUOP   ;;CPU TYPE,OPTIONS
                       ;*                      BITS 15-11=CPU TYPE
                       ;*                               11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                       ;*                               11/70=06,PDQ=07,Q=10
                       ;*                      BIT 10=REAL TIME CLOCK
                       ;*                      BIT  9=FLOATING POINT PROCESSOR
                       ;*                      BIT  8=MEMORY MANAGEMENT
001746                 $ETEND:
```

```
                                 .MEXIT
001746      000                  KB11E:   .BYTE   0        ;1174 WITHOUT MP CACHE FLAG
001747      000                  KB11EM:  .BYTE   0        ;1174 WITH MP CACHE FLAG
001750      000                  KB11CM:  .BYTE   0        ;KB11CM FLAG (1170 WITH MP MODS)
001751      000                  CISP:    .BYTE   0        ;CISP OPTION PRESENT FLAG

                                 ;OPCODE FOR MFPT INSTRUCTION (AVAILABLE ON KB11-E AND KB11-EM ONLY)
         000007                          MFPT=7
```

```
                                        .SBTTL  ERROR POINTER TABLE

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

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


001752                                  $ERRTB:


                                        ;ERROR TABLE FOR ERROR TYPE OUT:
                                        ;ITEM 1
001752  072022  111261  115044                  .WORD   EM1,DH1,DT1,DF1
001760  114521
                                        ;ITEM 2
001762  072107  111334  115056                  .WORD   EM2,DH2,DT2,DF2
001770  114525
                                        ;ITEM 3
001772  072301  111334  115056                  .WORD   EM3,DH3,DT3,DF3
002000  114525
                                        ;ITEM 4
002002  072415  111334  115056                  .WORD   EM4,DH4,DT4,DF4
002010  114525
                                        ;ITEM 5
002012  072530  111435  115074                  .WORD   EM5,DH5,DT5,DF5
002020  114533
                                        ;ITEM 6
002022  072610  111435  115074                  .WORD   EM6,DH6,DT6,DF6
002030  114533
                                        ;ITEM 7
002032  072670  111435  115074                  .WORD   EM7,DH7,DT7,DF7
002040  114533
                                        ;ITEM 10
002042  072762  111435  115074                  .WORD   EM10,DH10,DT10,DF10
002050  114533
                                        ;ITEM 11
002052  073053  111467  115116                  .WORD   EM11,DH11,DT11,DF11
002060  114543
                                        ;ITEM 12
002062  073141  111543  115142                  .WORD   EM12,DH12,DT12,DF12
002070  114554
                                        ;ITEM 13
002072  073266  111636  115160                  .WORD   EM13,DH13,DT13,DF13
002100  114562
                                        ;ITEM 14
002102  073346  111711  115174                  .WORD   EM14,DH14,DT14,DF14
002110  114567
                                        ;ITEM 15
002112  073405  112004  115210                  .WORD   EM15,DH15,DT15,DF15
```

```
002120  114574
                                  ;ITEM 16
002122  073455  112030  115216              .WORD   EM16,DH16,DT16,DF16
002130  114576
                                  ;ITEM 17
002132  073531  112116  115234              .WORD   EM17,DH17,DT17,DF17
002140  114604
                                  ;ITEM 20
002142  073531  112116  115316              .WORD   EM20,DH20,DT20,DF20
002150  114604
                                  ;ITEM 21
002152  073612  112176  115400              .WORD   EM21,DH21,DT21,DF21
002160  114634
                                  ;ITEM 22
002162  073676  112251  115456              .WORD   EM22,DH22,DT22,DF22
002170  114662
                                  ;ITEM 23
002172  074112  112320  115466              .WORD   EM23,DH23,DT23,DF23
002200  114665
                                  ;ITEM 24
002202  073676  112251  115502              .WORD   EM24,DH24,DT24,DF24
002210  114662
                                  ;ITEM 25
002212  074112  112320  115512              .WORD   EM25,DH25,DT25,DF25
002220  114665
                                  ;ITEM 26
002222  074246  112410  115526              .WORD   EM26,DH26,DT26,DF26
002230  114672
                                  ;ITEM 27
002232  074413  112455  115540              .WORD   EM27,DH27,DT27,DF27
002240  114676
                                  ;ITEM 30
002242  073676  112251  115554              .WORD   EM30,DH30,DT30,DF30
002250  114662
                                  ;ITEM 31
002252  074560  112320  115564              .WORD   EM31,DH31,DT31,DF31
002260  114665
                                  ;ITEM 32
002262  073676  112251  115600              .WORD   EM32,DH32,DT32,DF32
002270  114662
                                  ;ITEM 33
002272  074560  112320  115610              .WORD   EM33,DH33,DT33,DF33
002300  114665
                                  ;ITEM 34
002302  074717  112547  115624              .WORD   EM34,DH34,DT34,DF34
002310  114703
                                  ;ITEM 35
002312  075023  112547  115624              .WORD   EM35,DH35,DT35,DF35
002320  114703
                                  ;ITEM 36
002322  075132  112627  115640              .WORD   EM36,DH36,DT36,DF36
002330  114710
                                  ;ITEM 37
002332  075264  112674  115652              .WORD   EM37,DH37,DT37,DF37
002340  114714
                                  ;ITEM 40
```

```
002342  075346  113026  115700                  .WORD   EM40,DH40,DT40,DF40
002350  114726
                                        ;ITEM 41
002352  075521  112761  115666                  .WORD   EM41,DH41,DT41,DF41
002360  114722
                                        ;ITEM 42
002362  075705  112761  115666                  .WORD   EM42,DH42,DT42,DF42
002370  114722
                                        ;ITEM 43
002372  076160  113026  115700                  .WORD   EM43,DH43,DT43,DF43
002400  114726
                                        ;ITEM 44
002402  076306  113075  115722                  .WORD   EM44,DH44,DT44,DF44
002410  114736
                                        ;ITEM 45
002412  076502  113075  115722                  .WORD   EM45,DH45,DT45,DF45
002420  114736
                                        ;ITEM 46
002422  076701  113170  115762                  .WORD   EM46,DH46,DT46,DF46
002430  114755
                                        ;ITEM 47
002432  077022  113170  115762                  .WORD   EM47,DH47,DT47,DF47
002440  114755
                                        ;ITEM 50
002442  076306  113075  116014                  .WORD   EM50,DH50,DT50,DF50
002450  114736
                                        ;ITEM 51
002452  076502  113075  116014                  .WORD   EM51,DH51,DT51,DF51
002460  114736
                                        ;ITEM 52
002462  076701  113170  116054                  .WORD   EM52,DH52,DT52,DF52
002470  114755
                                        ;ITEM 53
002472  077022  113170  116054                  .WORD   EM53,DH53,DT53,DF53
002500  114755
                                        ;ITEM 54
002502  077146  113214  116106                  .WORD   EM54,DH54,DT54,DF54
002510  114771

                                        ;ITEM   55

002512  077321                                  EM55
002514  113255                                  DH55
002516  116116                                  DT55
002520  000000                                  0



                                        ;ITEM   56

002522  077352                                  EM56
002524  113255                                  DH55
002526  116116                                  DT55
002530  000000                                  0


```

```
                                   ;ITEM   57

002532  077420                             EM57
002534  113255                             DH55
002536  116116                             DT55
002540  114774                             DF57


                                   ;ITEM   60

002542  077461                             EM60
002544  113255                             DH55
002546  116116                             DT55
002550  114774                             DF61

                                   ;ITEM   61

002552  077520                             EM61
002554  113255                             DH55
002556  116116                             DT55
002560  114774                             DF61


                                   ;ITEM   62

002562  077607                             EM62
002564  113255                             DH55
002566  116116                             DT55
002570  114774                             DF62


                                   ;ITEM   63

002572  077641                             EM63
002574  113255                             DH55
002576  116116                             DT55
002600  114774                             DF63


                                   ;ITEM   64

002602  077717                             EM64
002604  113255                             DH55
002606  116116                             DT55
002610  114774                             DF64


                                   ;ITEM   65

002612  100000                             EM65
002614  113255                             DH55
002616  116116                             DT55
002620  114774                             DF65


                                   ;ITEM   66
```

```

002622  100066                         EM66
002624  113271                         DH66
002626  116124                         DT66
002630  114776                         DF66


                              ;ITEM    67

002632  100176                         EM67
002634  113271                         DH66
002636  116124                         DT66
002640  114776                         DF67


                              ;ITEM    70

002642  100311                         EM70
002644  113271                         DH66
002646  116124                         DT66
002650  114776                         DF70

                              ;ITEM    71

002652  100425                         EM71
002654  113335                         DH71
002656  116124                         DT66
002660  114776                         DF71

                              ;ITEM    72

002662  100473                         EM72
002664  113271                         DH66
002666  116124                         DT66
002670  114776                         DF72

                              ;ITEM    73

002672  100622                         EM73
002674  113271                         DH66
002676  116124                         DT66
002700  114776                         DF73

                              ;ITEM    74

002702  100732                         EM74
002704  113271                         DH66
002706  116124                         DT66
002710  114776                         DF74

                              ;ITEM    75

002712  101034                         EM75
002714  113271                         DH66
002716  116124                         DT66
002720  114776                         DF75
```

```

                                ;ITEM   76

002722  101150                          EM76
002724  113271                          DH66
002726  116124                          DT66
002730  114776                          DF76

                                ;ITEM   77

002732  101261                          EM77
002734  113271                          DH66
002736  116124                          DT66
002740  114776                          DF77

                                ;ITEM   0

002742  000000  000000  000000          .WORD   0,0,0,0
002750  000000

                                ;ITEM   0

002752  000000  000000  000000          .WORD   0,0,0,0
002760  000000

                                ;ITEM   0

002762  000000  000000  000000          .WORD   0,0,0,0
002770  000000

                                ;ITEM   103
002772  101524                   EM103  ;NO PARITY ERROR TRAP ON VALID STORE PARITY ERROR
002774  113255                   DH55
002776  116116                   DT55
003000  114774                   DF103

                                ;ITEM   104

003002  101605                   EM104  ;TEST-DATA-REFERENCE GIVING VALID STORE PARITY
003004  113255                   DH55   ;ERROR WAS NOT A MISS
003006  116116                   DT55
003010  114774                   DF104

                                ;ITEM   105

003012  101711                   EM105  ;FVPE DID NOT GET CLEARED AFTER VSPE OCCURED
003014  113255                   DH55
003016  116116                   DT55
003020  114774                   DF105

                                ;ITEM   106

003022  101765                   EM106  ;VALID-STORE-PARITY-ERROR BIT DID NOT SET IN CCR ONVSPE
003024  113255                   DH55
003026  116116                   DT55
003030  114774                   DF106
```

```

                                               ;ITEM   107

003032  102055                                  EM107  ;FAST ADDRESS MEMORY PARITY ERROR BITS (4,5) NOT
003034  113367                                  DH107  ;SET CORRECTLY IN MSER ON VSPE
003036  116154                                  DT107
003040  115010                                  DF107

                                               ;ITEM   110

003042  102174                                  EM110  ;VSIV SWITCHED ON VSPE
003044  113255                                  DH55
003046  116116                                  DT55
003050  114774                                  DF110

                                               ;ITEM   111

003052  102222                                  EM111  ;MEMORY SYSTEM ERROR REGISTER COULD NOT BE CLEARED
003054  113441                                  DH111
003056  116116                                  DT55
003060  114774                                  DF111

                                               ;ITEM   112

003062  102304                                  EM112  ;VSPE COULD NOT BE CLEARED IN CCR
003064  113255                                  DH55
003066  116116                                  DT55
003070  114774                                  DF112

                                               ;ITEM   113

003072  102345                                  EM113  ;TEST-DATA-REFERENCE NOT A HIT
003074  113271                                  DH66
003076  116124                                  DT66
003100  114776                                  DF113

                                               ;ITEM   0

003102  000000  000000  000000                         .WORD   0,0,0,0
003110  000000

                                               ;ITEM   115

003112  102403                                  EM115  ;TEST DATA REFERENCE NOT A MISS
003114  113456                                  DH115  ;CACHE DID NOT TURN OFF ON BACK-TO-BACK FLUSH
003116  116116                                  DT55
003120  114774                                  DF115

                                               ;ITEM   116

003122  000000  000000  000000                         .WORD   0,0,0,0
003130  000000

                                               ;ITEM   117

003132  000000  000000  000000                         .WORD   0,0,0,0
```

```
003140  000000

                                        ;ITEM   120

003142  000000  000000  000000                  .WORD   0,0,0,0
003150  000000

                                        ;ITEM   121

003152  000000  000000  000000                  .WORD   0,0,0,0
003160  000000

                                        ;ITEM   122

003162  000000  000000  00000C                  .WORD   0,0,0,0
003170  000000

                                        ;ITEM 123
003172  102520                          EM123                   ;BYP BIT IN KIPDR COULD NOT BE CLEARED
003174  113475                          DH123                   ;     PC    KIPDR   (KIPDR)
003176  116170                          DT123                   ;$ERRPC,$REG0,$REG1,0
003200  115015                          DF123                   ;0,0,0
                                        ;ITEM 124
003202  102566                          EM124                   ;BYP BIT IN KIPDR COULD NOT BE SET
003204  113475                          DH123
003206  116170                          DT123
003210  115015                          DF123

                                        ;ITEM 125
003212  102630                          EM125                   ;TEST DATA COULD NOT BE MADE HIT
003214  113524                          DH125                   ; PC   CCR   PARADR  PAR  PDR   TST-DATA-ADR
003216  116200                          DT125                   ;$ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,0
003220  115002                          DF100

                                        ;ITEM 126
003222  102670                          EM126                   ;TEST DATA REFERENCE NOT A MISS
                                                                ;CACHED DATA WAS NOT FORCED A MISS ON VIRTUAL PAGE BYPASS
003224  113524                          DH125
003226  116200                          DT125
003230  115002                          DF100

                                        ;ITEM 127
003232  103021                          EM127                   ;TEST DATA REFERENCE NOT A MISS
                                                                ;CACHED DATA WAS NOT INVALIDATED ON VIRTUAL BYPASS
003234  113524                          DH125
003236  116200                          DT125
003240  115002                          DF100
                                        ;ITEM 130
003242  103150                          EM130                   ;BYP BIT IN SIPDR COULD NOT BE CLEARED
003244  113613                          DH130                   ;     PC    SIPDR    (SIPDR)
003246  116170                          DT123
003250  115015                          DF123

                                        ;ITEM 131
003252  103216                          EM131                   ;BYP IN SIPDR COULD NOT BE SET
003254  113613                          DH130
```

```
003256  116170                    DT123
003260  115015                    DF123

                                  ;ITEM 132

003262  103260                    EM132                      ;BYP BIT IN UIPDR COULD NOT BE CLEARED
003264  113642                    DH132                      ;   PC      UIPDR     (UIPDR)
003266  116170                    DT123
003270  115015                    DF123

                                  ;ITEM 133
003272  103326                    EM133                      ;BYP BIT IN UIPDR COULD NOT BE SET
003274  113642                    DH132
003276  116170                    DT123
003300  115015                    DF123
                                  ;ITEM 0
003302  000000  000000  000000            .WORD    0,0,0,0
003310  000000
                                  ;ITEM 0
003312  000000  000000  000000            .WORD    0,0,0,0
003320  000000
                                  ;ITEM 136
003322  103370  113671  116216            .WORD    EM136,DH136,DT136,DF136
003330  115020
                                  ;ITEM 137
003332  103605  113671  116216            .WORD    EM137,DH137,DT137,DF137
003340  115020
                                  ;ITEM 140
003342  104023  113736  116230            .WORD    EM140,DH140,DT140,DF140
003350  115024
                                  ;ITEM 141
003352  104364  113736  116230            .WORD    EM141,DH141,DT141,DF141
003360  115024
                                  ;ITEM 142
003362  104724  113736  116230            .WORD    EM142,DH142,DT142,DF142
003370  115024
                                  ;ITEM 143
003372  105266  113736  116230            .WORD    EM143,DH143,DT143,DF143
003400  115024
                                  ;ITEM 144
003402  105627  113736  116230            .WORD    EM144,DH144,DT144,DF144
003410  115024
                                  ;ITEM 145
003412  106161  113736  116230            .WORD    EM145,DH145,DT145,DF145
003420  115024
                                  ;ITEM 146
003422  106512  113736  116230            .WORD    EM146,DH146,DT146,DF146
003430  115024
                                  ;ITEM 147
003432  107045  113736  116230            .WORD    EM147,DH147,DT147,DF147
003440  115024
                                  ;ITEM 150
003442  107377  114001  116242            .WORD    EM150,DH150,DT150,DF150
003450  115030
                                  ;ITEM 151
003 `2  107463  114065  116254            .WORD    EM151,DH151,DT151,DF151
```

```
003460  115034
                                 ;ITEM 152
003462  107463  114134  116254            .WORD   EM152,DH152,DT152,DF152
003470  115034
                                 ;ITEM 153
003472  107463  114203  116254            .WORD   EM153,DH153,DT153,DF153
003500  115034
                                 ;ITEM 154
003502  107544  114265  116264            .WORD   EM154,DH154,DT154,DF154
003510  115037
                                 ;ITEM 155
003512  107576  114323  116264            .WORD   EM155,DH155,DT155,DF155
003520  115037
                                 ;ITEM 156
003522  107630  114361  116264            .WORD   EM156,DH156,DT156,DF156
003530  115037
                                 ;ITEM 0
003532  000000  000000  000000            .WORD   0,0,0,0
003540  000000
                                 ;ITEM 160
003542  107675  114417  116254            .WORD   EM160,DH160,DT160,DF160
003550  115037
                                 ;ITEM 161
003552  107727  114445  116254            .WORD   EM161,DH161,DT161,DF161
003560  115037

                                 ;ITEM 162
003562  107775  114475  116276            .WORD   EM162,DH162,DT162,DF55
003570  114774

                                 ;ITEM 163
003572  110205  114475  116276            .WORD   EM163,DH162,DT162,DF55
003600  114774

                                 ;ITEM 164
003602  110277  114475  116276            .WORD   EM164,DH162,DT162,DF55
003610  114774

                                 ;ITEM 165
003612  110356  114475  116276            .WORD   EM165,DH162,DT162,DF55
003620  114774

                                 ;ITEM 166
003622  110375  114475  116276            .WORD   EM166,DH162,DT162,DF55
003630  114774

                                 ;ITEM 167
003632  110461  114475  116276            .WORD   EM167,DH162,DT162,DF55
003640  114774

                                 ;ITEM 170
003642  110565  114475  116276            .WORD   EM170,DH162,DT162,DF55
003650  114774

                                 ;ITEM 171
003652  110630  114475  116276            .WORD   EM171,DH162,DT162,DF55
```

```
003660  114774

                                        ;ITEM 172
003662  110674  114475  116276                  .WORD   EM172,DH162,DT162,DF55
003670  114774

                                        ;ITEM 173
003672  110760  114475  116276                  .WORD   EM173,DH162,DT162,DF55
003700  114774

                                        ;ITEM 174
003702  111146  114475  116276                  .WORD   EM174,DH162,DT162,DF55
003710  114774

                                        ;ITEM 175
003712  111211  114475  116276                  .WORD   EM175,DH162,DT162,DF55
003720  114774
003722  000016                          RS4REG: .WORD   16
003724  172040                          RS4CS1: .WORD   172040
003726  000000                          RS4WC:  .WORD   0
003730  000000                          RS4BA:  .WORD   0
003732  000000                          RS4DA:  .WORD   0
003734  000000                          RS4CS2: .WORD   0
003736  000000                          RS4DS:  .WORD   0
003740  000000                          RS4ER:  .WORD   0
003742  000000                          RS4AS:  .WORD   0
003744  000000                          RS4LA:  .WORD   0
003746  000000                          RS4DB:  .WORD   0
003750  000000                          RS4MR:  .WORD   0
003752  000000                          RS4DT:  .WORD   0
003754  000000                          RS4BAE: .WORD   0
003756  000000                          RS4CS3: .WORD   0

003760  000026                          RP4REG: .WORD   26
003762  176700                          RP4CS1: .WORD   176700
003764  000000                          RP4WC:  .WORD   0
003766  000000                          RP4BA:  .WORD   0
003770  000000                          RP4DA:  .WORD   0
003772  000000                          RP4CS2: .WORD   0
003774  000000                          RP4DS:  .WORD   0
003776  000000                          RP4RR1: .WORD   0
004000  000000                          RP4AS:  .WORD   0
004002  000000                          RP4LA:  .WORD   0
004004  000000                          RP4DB:  .WORD   0
004006  000000                          RP4MR:  .WORD   0
004010  000000                          RP4DT:  .WORD   0
004012  000000                          RP4SN:  .WORD   0
004014  000000                          RP4OF:  .WORD   0
004016  000000                          RP4DC:  .WORD   0
004020  000000                          RP4CCC: .WORD   0
004022  000000                          RP4RR2: .WORD   0
004024  000000                          RP4RR3: .WORD   0
004026  000000                          RP4EC1: .WORD   0
004030  000000                          RP4EC2: .WORD   0
004032  000000                          RP4BAE: .WORD   0
004034  000000                          RP4CS3: .WORD   0
```

```

004036 000014                      RH4REG: .WORD   14
004040 160100                      RH4CS1: .WORD   160100
004042 000000                      RH4WC:  .WORD   0
004044 000000                      RH4BA:  .WORD   0
004046 000000                      RH4MR2: .WORD   0
004050 000000                      RH4CS2: .WORD   0
004052 000000                      RH4ST:  .WORD   0
004054 000000                      RH4ER:  .WORD   0
004056 000000                      RH4AS:  .WORD   0
004060 000000                      RH4DR:  .WORD   0
004062 000000                      RH4DB:  .WORD   0
004064 000000                      RH4MR1: .WORD   0
004066 000000                      RH4DT:  .WORD   0

004070 000002                      RH4REX: .WORD   2
004072 160174                      RH4AE:  .WORD   160174
004074 000000                      RH4CS3: .WORD   0

004076 000007                      RK5REG: .WORD   7
004100 177400                      RK5DS:  .WORD   177400
004102 000000                      RK5ER:  .WORD   0
004104 000000                      RK5CS1: .WORD   0
004106 000000                      RK5WC:  .WORD   0
004110 000000                      RK5BA:  .WORD   0
004112 000000                      RK5DA:  .WORD   0
004114 000000                      RK5DB:  .WORD   0


004116 000006                      UBEREG: .WORD   6
004120 170000                      UBEDB:  .WORD   170000
004122 000000                      UBECC:  .WORD   0
004124 000000                      UBEBA:  .WORD   0
004126 000000                      UBECR1: .WORD   0
004130 000000                      UBECLR: .WORD   0
004132 000000                      UBECR2: .WORD   0

                                   ;THESE ARE THE DEVICE TRAP VECTER ADDRESSES:
004134 000204                      RS4V:   .WORD   204
004136 000254                      RP4V:   .WORD   254
004140 000774                      RH4V:   .WORD   774
004142 000220                      RK5V:   .WORD   220
004144 000510                      UBEV:   .WORD   510



004146 005037 001502               START:  CLR     $TSTNM
                                   .SBTTL  INITIALIZE THE COMMON TAGS
                                   ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
004152 012706 001500                       MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
004156 005026                              CLR     (R6)+           ;;CLEAR MEMORY LOCATION
004160 022706 001540                       CMP     #SWR,R6 ;;DONE?
004164 001374                              BNE     .-6             ;;LOOP BACK IF NO
004166 012706 001400                       MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                   ;;INITIALIZE A FEW VECTORS
004172 012737 051520 000020                MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
```

```
004200  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
004206  012737  052006  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
004214  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
004222  012737  054374  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
004230  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
004236  012737  054516  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
004244  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
004252  013737  051414  051406          MOV     $ENDCT,$EOPCT    ;;SETUP END-OF-PROGRAM COUNTER
004260  005037  001702                  CLR     $TIMES           ;;INITIALIZE NUMBER OF ITERATIONS
004264  005037  001704                  CLR     $ESCAPE          ;;CLEAR THE ESCAPE ON ERROR ADDRESS
004270  112737  000001  001515          MOVB    #1,$ERMAX        ;;ALLOW ONE ERROR PER TEST
004276  012737  004276  001506          MOV     #.,$LPADR        ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
004304  012737  004304  001510          MOV     #.,$LPERR        ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A ''-1'', SETUP FOR A SOFTWARE SWITCH REGISTER.
004312  013746  000004                  MOV     @#ERRVEC,-(SP)   ;;SAVE ERROR VECTOR
004316  012737  004352  000004          MOV     #64$,@#ERRVEC    ;;SET UP ERROR VECTOR
004324  012737  177570  001540          MOV     #DSWR,SWR        ;;SETUP FOR A HARDWARE SWICH REGISTER
004332  012737  177570  001542          MOV     #DDISP,DISPLAY   ;;AND A HARDWARE DISPLAY REGISTER
004340  022777  177777  175172          CMP     #-1,@SWR         ;;TRY TO REFERENCE HARDWARE SWR
004346  001012                          BNE     66$              ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                 ;;AND  THE HARDWARE SWR IS NOT = -1
004350  000403                          BR      65$              ;;BRANCH IF NO TIMEOUT
004352  012716  004360          64$:    MOV     #65$,(SP)        ;;SET UP FOR TRAP RETURN
004356  000002                          RTI
004360  012737  000176  001540  65$:    MOV     #SWREG,SWR       ;;POINT TO SOFTWARE SWR
004366  012737  000174  001542          MOV     #DISPREG,DISPLAY
004374  012637  000004          66$:    MOV     (SP)+,@#ERRVEC   ;;RESTORE ERROR VECTOR

004400  005037  001724                  CLR     $PASS            ;;CLEAR PASS COUNT
004404  132737  000200  001737          BITB    #APTSIZE,$ENVM   ;;TEST USER SIZE UNDER APT
004412  001403                          BEQ     67$              ;;YES,USE NON-APT SWITCH
004414  012737  001740  001540          MOV     #$SWREG,SWR      ;;NO,USE APT SWITCH REGISTER
004422                          67$:
                                .SBTTL  TYPE PROGRAM NAME
                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
004422  005227  177777                  INC     #-1              ;;FIRST TIME?
004426  001046                          BNE     68$              ;;BRANCH IF NO
004430  022737  051464  000042          CMP     #$ENDAD,@#42     ;;ACT-11?
004436  001442                          BEQ     68$              ;;BRANCH IF YES
004440  104401  004506                  TYPE    ,69$             ;;TYPE ASCIZ STRING
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
004444  005737  000042                  TST     @#42             ;;ARE WE RUNNING UNDEK XXDP/ACT?
004450  001012                          BNE     70$              ;;BRANCH IF YES
004452  123727  001736  000001          CMPB    $ENV,#1          ;;ARE WE RUNNING UNDER APT?
004460  001406                          BEQ     70$              ;;BRANCH IF YES
004462  023727  001540  000176          CMP     SWR,#SWREG       ;;SOFTWARE SWITCH REG SELECTED?
004470  001005                          BNE     71$              ;;BRANCH IF NO
004472  104406                          GTSWR                    ;;GET SOFT-SWR SETTINGS
004474  000403                          BR      71$
004476  112737  000001  001534  70$:    MOVB    #1,$AUTOB        ;;SET AUTO-MODE INDICATOR
004504                          71$:
004504  000417                          BR      68$              ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <CRLF>'CEKBD-E   11/70 CACHE  #2 '<CRLF>
004544                          68$:
004544  005227  177777                  INC     #-1              ;TYPE MESSAGE FIRST PASS ONLY
```

```
004550  001043                                  BNE     72$
004552  022737  051464  000042                  CMP     #$ENDAD,@#42    ;SKIP IF ACT
004560  001437                                  BEQ     72$
004562  104401  004570                          TYPE    ,73$
004566  000434                                  BR      72$
004570  050200  047522  051107  73$:    .ASCIZ  <CRLF>'PROGRAMMABLE RP04 DRIVES WILL NOT BE USED BY TEST 35'<CRLF>
004576  046501  040515  046102
004604  020105  050122  032060
004612  042040  044522  042526
004620  020123  044527  046114
004626  047040  052117  041040
004634  020105  051525  042105
004642  041040  020131  042524
004650  052123  031440  100065
004656     000
        004660                                  .EVEN
004660                          72$:
                                ;;*****************************************************************

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

004660  052737  000200  054752                  BIS     #BIT07,$KT11
004666  004737  054670                          JSR     PC,$SIZE
004672  062737  000037  055270                  ADD     #37,$LSTBK              ;ADJUST THE SIZE FOR COMPARISON
                                                                                ;TO SIZE REGISTER
004700  023737  177760  055270                  CMP     @#SIZELO,$LSTBK         ;IS THE ACTUAL SIZE REFLECTED BY THE
                                                                                ;SIZE REGISTER?
004706  001420                                  BEQ     OKSIZ
004710  104401  071411                          TYPE    ,MS01
004714  104401  071546                          TYPE    ,MS02
004720  104401  001713                          TYPE    ,$CRLF
004724  013746  177760                          MOV     @#SIZELO,-(SP)  ;;SAVE @#SIZELO FOR TYPEOUT
004730  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
004732     006                                  .BYTE   6               ;;TYPE 6 DIGIT(S)
004733     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
004734  104401  071575                          TYPE    ,MS03
004740  013746  055270                          MOV     $LSTBK,-(SP)    ;;SAVE $LSTBK FOR TYPEOUT
004744  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
004746     006                                  .BYTE   6               ;;TYPE 6 DIGIT(S)
004747     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS
004750                          OKSIZ:

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

```
004750  105037  001750          KBTST:  CLRB    @#KB11CM         ;RESET THE MP FLAG
004754  005037  001746                  CLR     @#KB11E          ;CLEAR KB11E AND KB11EM FLAGS
004760  012737  005216  000010          MOV     #MFPTTR,@#RESVEC ;SET UP TRAP ADDRESS FOR MFPT AT RESERV VECTOR
004766  000007                          MFPT                     ;EXECUTE MFPT. WILL TRAP ON 1170 (KB11B/C) OR
                                                                 ;KB11CM (11/74          )
004770  012737  000001  001746          MOV     #1,@#KB11E       ;HERE IF KB11E OR KB11EM. SET FLAG
004776  005037  177750          T1:     CLR     @#MAINT          ;CLEAR THE MAINTENANCE REGISTER
005002  005005                          CLR     R5               ;RESET THE TEST COUNTER
005004  012700  177746                  MOV     #CONTRL,R0       ;GET THE ADDRESS OF...
005010  012701  177750                  MOV     #MAINT,R1        ;CCR,MAINT,AND MAPH00...
005014  012702  170202                  MOV     #MAPH00,R2       ;AND PLACE IN R0-R2
005020  052710  040000                  BIS     #BIT14,(R0)      ;TRY TO SET IVSS BIT
005024  032710  040000                  BIT     #BIT14,(R0)      ;DID IT SET?
005030  001403                          BEQ     T2               ;NO,GO TO NEXT TEST
005032  042710  040000                  BIC     #BIT14,(R0)      ;CLEAR IT.
005036  005205                          INC     R5               ;TEST IS POSITIVE
005040  052711  000001          T2:     BIS     #BIT0,(R1)       ;SET EDMA IN MAINT REGISTER
005044  032711  000001                  BIT     #BIT0,(R1)
005050  001410                          BEQ     T3
005052  052710  004000                  BIS     #BIT11,(R0)      ;TRY TO SET DMMA IN CCR
005056  032710  004000                  BIT     #BIT11,(R0)
005062  001403                          BEQ     T3
005064  042710  004000                  BIC     #BIT11,(R0)
005070  005205                          INC     R5
005072  042711  000001          T3:     BIC     #BIT0,(R1)       ;MAKE SURE EDMA IS CLEAR
005076  052737  100000  172300          BIS     #BIT15,KIPDR0    ;TRY TO SET BYP ON A PDR
005104  032737  100000  172300          BIT     #BIT15,KIPDR0
005112  001404                          BEQ     T4
005114  042737  100000  172300          BIC     #BIT15,KIPDR0
005122  005205                          INC     R5
005124  052712  100000          T4:     BIS     #BIT15,(R2)      ;TRY TO SET BYP ON UNIBUS MAP
005130  032712  100000                  BIT     #BIT15,(R2)
005134  001403                          BEQ     T.END
005136  042712  100000                  BIC     #BIT15,(R2)
005142  005205                          INC     R5
005144  022705  000002          T.END:  CMP     #2,R5            ;IS THE RESULT OF THE TEST >=2
005150  101021                          BHI     2$               ;NO,THIS IT A KB11E OR KB11-B/C (11/70)
005152  005000                          CLR     R0
005154  005037  177746                  CLR     @#CONTRL         ;CLEAR CACHE CONT. REG. AND
005160  013701  177746          3$:     MOV     @#CONTRL,R1      ;WAIT UNTILL VCIP BIT CLEARS
005164  001402                          BEQ     4$               ;OR THE COUNT RUNS OUT
005166  005200                          INC     R0
005170  001373                          BNE     3$
005172  005737  001746          4$:     TST     @#KB11E          ;IS IS A KB11-E OR KB11-EM?
005176  001404                          BEQ     1$               ;BR IF NEITHER. MUST BE KB11CM
005200  012737  000400  001746          MOV     #BIT8,@#KB11E    ;SET UPPER BYTE (KB11-EM)
005206  000402                          BR      2$               ;DONE
005210  105237  001750          1$:     INCB    @#KB11CM         ;YES, FLAG THIS AS A MODIFIED PROCESSOR
005214  000403                  2$:     BR      ENDKB            ;DONE DETERMINING WHICH CPU

005216                          MFPTTR:                          ;HERE IF MFPT TRAPPED. SEE IF 1170 OR KB11CM
005216  012716  004776                  MOV     #T1,(SP)         ;SET UP RETURN ADDRESS FOR RTI
005222  000002                          RTI                      ;RETURN
005224                          ENDKB:
005224  005227  177777                  INC     #-1              ;FIRST TIME?
005230  001026                          BNE     100$             ;BR IF NO
```

```
005232 104401 071605                     TYPE    ,MSG1            ;<15><12>CPU UNDER TEST FOUND TO BE A
005236 005737 001746                     TST     @#KB11E          ;IS THIS A KB11-E OR KB11-EM?
005242 001011                            BNE     101$             ;BR IF EITHER ONE
005244 105737 001750                     TSTB    @#KB11CM         ;IS IT A 11/74         (KB11CM)
005250 001003                            BNE     1$               ;BR IF IT IS
005252 104401 071655                     TYPE    ,MSG3            ;KB11-B/C<15><12>
005256 000413                            BR      100$             ;SKIP OTHER MESSAGE
005260 104401 071667             1$:     TYPE    ,MSG4            ;11/74         (KB11CM)<15><12>
005264 000410                            BR      100$             ;SKIP CISP MESSAGE
005266 105737 001746             101$:   TSTB    @#KB11E          ;IS IT A KB11-E?
005272 001403                            BEQ     102$             ;BR IF NOT. MUST BE KB11-EM
005274 104401 071720                     TYPE    ,MSG5            ;KB11-E<15><12>
005300 000402                            BR      100$             ;SKIP KB11-EM MESSAGE
005302 104401 071644             102$:   TYPE    ,MSG2            ;KB11-EM<15><12>
005306                           100$:

                                 ;THIS ROUTINF SAVES THE TOP 1500 (DEC) WORDS OF THE FIRST 28K OF
                                 ;MEMORY. THESE LOCATIONS SHOULD CONTAIN EITHER THE MONITOR OR THE
                                 ;LOADER WHICH LOADED THE PROGRAM. NOTE THAT TO RESTORE THIS PART
                                 ;OF CORE, THAT IS TO RESTORE THE LOADER OR MONITOR, ALL THE USER
                                 ;MUST DO IS TYPE ^C (CONTROL-C), WHILE THIS PROGRAM IS RUNNING.
                                 ;THIS WILL AUTOMATICALLY RESTORE THE TOP PART OF MEMORY TO ITS STATE
                                 ;BEFORE THIS PROGRAM WAS STARTED! AFTER THE MONITOR (OR LOADER) HAS BEEN
                                 ;RESTORED THIS PROGRAM WILL HALT.
005306 005237 056254             LOOP:   INC     MONF             ;INCREMENT THE FLAG WHICH INDICATES
005312 001013                            BNE     TOP              ;WHETHER OR NOT THE TOP OF MEMORY
                                                                  ;IN THE FIRST 28K HAS BEEN SAVED.
005314 013737 000060 056252              MOV     @#TKVEC,MONTTY   ;SAVE THE INITIAL CONTENTS OF THE TTY
                                                                  ;KEYBOARD INTERRUPT VECTOR.
005322 012700 002734                     MOV     #^D1500,R0       ;IF NOT THEN SAVE IT.
005326 012701 120314                     MOV     #BOTTOM+4,R1     ;SAVE IT AT THE BOTTOM OF THIS PROGRAM.
005332 012702 160000                     MOV     #160000,R2       ;GET THE ADDRESS OF THE END OF THE MONITOR.
005336 014221                    1$:     MOV     -(R2),(R1)+      ;SAVE 1500 (DEC) LOCATIONS (WORDS)
005340 077002                            SOB     R0,1$
005342 012737 000044 177770      TOP:    MOV     #44,@#177770

005350 012737 056054 000060              MOV     #RESMON,@#TKVEC  ;SET THE KEYBOARD INTERRUPT VECTOR.
005356 012737 000340 000062              MOV     #340,@#TKVEC+2
005364 005077 174156                     CLR     @$TKB            ;MAKE SURE THE KEYBOARD BUFFER IS CLEAR.
005370 152777 000100 174146              BISB    #BIT6,@$TKS      ;TURN ON INTERRUPT ENABLE FOR THE KEYBOARD.
005376 012737 055412 000004              MOV     #CPSPUR,@#4      ;SET UP FOR UNEXPECTED ERRORS.
005404 012737 055440 000114              MOV     #SPUR,@#114

                                 ;;***********************************************************************
                                 ;*TEST 1           PARITY ERROR ABORT
                                 ;*
                                 ;*      THIS TEST ENSURES THAT A CACHE PARITY ERROR FLAG CAUSES AN ABORT.
                                 ;*      THIS IS DONE BY FORCING A PARITY ERROR ON AN EVEN WORD.
                                 ;;***********************************************************************
005412 000004                    TST1:   SCOPE
005414 012737 005570 001704              MOV     #2$,$ESCAPE      ;SETUP ESCAPE ADDRESS
005422 012737 000014 177746              MOV     #14,@#CONTRL     ;ENSURE MISSES TO BOTH GROUPS
005430 012737 005476 001510              MOV     #7$,$LPERR       ;SETUP ERROR LOOP
005436 012737 005570 000114              MOV     #2$,@#CACHVEC    ;SETUP CACHE VECTOR
005444 012737 005524 000004              MOV     #3$,@#ERRVEC     ;SETUP LOCATION 4
005452 012737 005540 000014              MOV     #4$,@#14         ;SETUP LOCATION 14
```

```
005460  012737  005554  000104          MOV     #5$,@#104         ;SETUP LOCATION 104
005466  012704  170000                  MOV     #170000,R4        ;PUT MAINTENANCE DATA IN R4
005472  012702  177750                  MOV     #MAINT,R2         ;PUT ADDRESS OF MAIN REG IN R2
005476  012706  001400          7$:     MOV     #STACK,SP         ;INITIALIZE THE SP
005502  000402                          BR      1$                ;GO TO NEXT INSTRUCTION
        005504                          LOC=.                     ;THIS IS
        005504                          LOC=-3&LOC                ;USED TO MAKE
        005510                          LOC=LOC+4                 ;1$ FALL ON
        005510                          .=LOC                     ;AND EVEN WORD
005510  000240                  1$:     NOP                       ;USED TO MAKE BAD PARITY INSTR ON EVEN WORD
005512  010412                          MOV     R4,(R2)           ;SET BITS IN MAINT REG
005514  005701                          TST     R1                ;EXECUTE INSTR TO CAUSE PE ABORT
                                ;FAILURE, NO ABORT
005516  005012                          CLR     (R2)              ;CLEAR MAINT REG
005520  000240                          NOP
005522  104162                          ERROR   162               ;NO PE ABORT
                                ;FAILURE, ABORTED TO WRONG VECTOR
005524  005012                  3$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005526  000240                          NOP
005530  012737  177777  177744          MOV     #-1,@#MEMERR
005536  104163                          ERROR   163               ;ABORTED TO LOCATION 4
005540  005012                  4$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005542  000240                          NOP
005544  012737  177777  177744          MOV     #-1,@#MEMERR
005552  104164                          ERROR   164               ;ABORTED TO 14
005554  005012                  5$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005556  000240                          NOP
005560  012737  177777  177744          MOV     #-1,@#MEMERR
005566  104165                          ERROR   165               ;ABORTED TO 104
                                ;TEST OK
005570  005012                  2$:     CLR     (R2)              ;ENSURE MAINT REG CLEAR
005572  000240                          NOP
005574  012737  177777  177744          MOV     #-1,@#MEMERR      ;CLEAR MEMORY ERROR REG
005602  012737  055412  000004          MOV     #CPSPUR,@#ERRVEC  ;RESET LOCATION 4
                                                                  ;CONTINUE
                                ;;*****************************************************************
                                ;*TEST 2        PARITY ERROR TRAP
                                ;*
                                ;*      THIS TEST ENSURES THAT A PARITY TRAP FUNCTIONS PROPERLY.
                                ;*      THIS IS DONE BY MAKING THE ODD WORD HAVE BAD PARITY.
                                ;*      IF THE TRAP DOESN'T OCCUR THEN THE PROBLEM IS ON TMCA.
                                ;*      IF A TRAP OCCURS TO THE WRONG VECTOR THE PROBLEM COULD BE
                                ;*      ON TMCA OR UBCB.
                                ;;*****************************************************************
005610  000004                  TST2:   SCOPE
005612  012737  005716  001704          MOV     #3$,$ESCAPE       ;SETUP ESCAPE ADDRESS
005620  012737  005652  001510          MOV     #1$,$LPERR        ;SETUP ERROR LOOP
005626  012737  005702  000004          MOV     #2$,@#ERRVEC      ;SETUP THE ERROR VECTOR
005634  012737  005716  000114          MOV     #3$,@#CACHVEC     ;SETUP THE CACHE VECTOR
005642  012704  170000                  MOV     #170000,R4        ;PUT MAINT DATA IN R4
005646  012702  177750                  MOV     #MAINT,R2         ;PUT MAINT REG ADDR IN R2
005652  012706  001400          1$:     MOV     #STACK,SP         ;INITIALIZE THE SP
005656  000402                          BR      4$                ;GO TO NEXT INSTRUCTION
        005660                          LOC=.                     ;THIS IS USED
        005660                          LOC=-3&LOC                ;TO MAKE
        005664                          LOC=LOC+4                 ;1$ FALL ON
        005664                          .=LOC                           ;AN EVEN WORD
005664  000240                  4$:     NOP                             ;GOOD PARITY ON EVEN WORD
005666  010412                          MOV     R4,(R2)                 ;SET BITS IN MAINT REG
005670  000240                          NOP
005672  005701                          TST     R1
                                ;FAILURE, NO TRAP
005674  005012                          CLR     (R2)                    ;ENSURE MAINT REG CLEAR
005676  000240                          NOP
005700  104166                          ERROR   166                     ;NO PE TRAP
                                ;FAILURE, TRAPPED TO WRONG VECTOR
005702  005012                  2$:     CLR     (R2)                    ;ENSURE MAINT REG CLEAR
005704  000240                          NOP
005706  012737  177777  177744          MOV     #-1,@#MEMERR            ;CLEAR MEM ERROR REG
005714  104167                          ERROR   167                     ;PE TRAP, TRAPPED TO
                                ;TEST OK
005716  005012                  3$:     CLR     (R2)                    ;ENSURE MAINT REG CLEAR
005720  000240                          NOP
005722  012737  177777  177744          MOV     #-1,@#MEMERR            ;CLEAR MEM ERROR REG
005730  012737  055412  000004          MOV     #CPSPUR,@#ERRVEC ;RESTORE LOCATION 4

005736  012737  055440  000250          MOV     #SPUR,@#MMVEC ;RESTORE MEM VEC
                                                                        ;CONTINUE
                                ;;*****************************************************************
                                ;*TEST 3                 MEM MGT AND PE TRAP PRIORITY ARBITRATION
                                ;*
                                ;*       THIS TEST ENSURES THAT THE ARBITRATION LOGIC WORKS FOR MEMORY
                                ;*       MANAGEMENT AND PARITY ERROR TRAPS.
                                ;*
                                ;;*****************************************************************
005744  000004                  TST3:   SCOPE
005746  012737  001400  172354  1$:     MOV     #1400,@#KIPAR6  ;RESTORE PAR6
005754  112737  000004  172314          MOVB    #4,@#KIPDR6     ;SETUP PAGE 6 TO TRAP ON ALL ACCESSES
005762  012704  170000                  MOV     #170000,R4      ;PUT MAINT REG DATA IN R4
005766  012702  177750                  MOV     #MAINT,R2       ;PUT ADDRESS OF MAINT REG IN R2
                                ;;*****************************************************************
                                ;PIR6 DISABLED BY MGMT

005772  012737  040000  140000          MOV     #BIT14,@#140000 ;PUT PIR6 ENABLE BIT IN PAGE 6
006000  012737  006052  000240          MOV     #3$,@#PIRQVEC   ;SETUP PIRQ VECTOR
006006  012737  000340  000252          MOV     #PR7,@#MMVEC+2  ;SET UP MMVEC PSW
006014  012737  006064  000250          MOV     #4$,@#MMVEC     ;SETUP MEM MGMT VECTOR
006022  012737  006030  001510          MOV     #5$,$LPERR      ;SETUP ERROR LOOP
006030  012706  001400          5$:     MOV     #STACK,SP       ;INITIALIZE THE SP
006034  012737  001001  177572          MOV     #1001,@#MMR0    ;TURN RELOCATION ON
006042  000235                          SPL     5               ;SET PROCESSOR AT LEVEL 5
006044  013737  140000  177772          MOV     @#140000,@#PIRQ ;SET PIR6 AND MEM MGT TRAP
                                ;FAILURE, PRI6 CAME THRU
006052  005037  177572          3$:     CLR     @#MMR0          ;TURN RELOCATION OFF
006056  005037  177772                  CLR     @#PIRQ          ;CLEAR PIR6
006062  104170                          ERROR   170     ;PIR6 CAME IN ON
                                ;
                                ;;*****************************************************************
                                ;PIR3 DISABLED BY MGMT
006064  005037  177572          4$:     CLR     @#MMR0          ;TURN RELOCATION OFF
006070  005037  177772                  CLR     @#PIRQ          ;CLEAR PIR LEVEL 6
006074  012737  006146  000240          MOV     #6$,@#PIRQVEC   ;SETUP PIRQ VECTOR
```

```
006102  012737  006160  000250            MOV     #7$,@#MMVEC     ;SETUP MEM MGT VECTOR
006110  012737  006124  001510            MOV     #8$,$LPERR      ;SETUP ERROR LOOP
006116  012737  004000  140000            MOV     #BIT11,@#140000 ;PUT PIR3 ENABLE BIT IN PAGE 6
006124  012706  001400          8$:       MOV     #STACK,SP       ;INITIALIZE THE SP
006130  012737  001001  177572            MOV     #1001,@#MMR0    ;TURN ON RELOCATION
006136  000232                            SPL     2               ;LOWER CPU TO LEVEL 2
006140  013737  140000  177772            MOV     @#140000,@#PIRQ ;SET PIR3 & MGMT
                                ;FAILURE, PIR3 CAME THRU
006146  005037  177572          6$:       CLR     @#MMR0          ;TURN OFF RELOCATION
006152  005037  177772                    CLR     @#PIRQ          ;CLEAR PIR3
006156  104171                            ERROR   171             ;PIR3 CAME IN ON
                                ;
                                ;;**************************************************************
                                ;STACK LIMIT YELLOW DISABLED BY PARITY ERROR
006160  005037  177572          7$:       CLR     @#MMR0          ;TURN RELOCATION OFF
006164  005037  177772                    CLR     @#PIRQ          ;CLEAR PIR LEVEL 3
006170  012737  006230  001510            MOV     #9$,$LPERR      ;SETUP ERROR LOOP
006176  012737  006254  000004            MOV     #10$,@#ERRVEC   ;SETUP THE ERROR VECTOR
006204  012737  006254  000114            MOV     #10$,@#CACHVEC  ;SETUP CACHEVEC
006212  012737  000240  000116            MOV     #PR5,@#CACHVEC+2 ;PUT PRIORITY 5 IN CACHE VECTOR PSW
006220  012704  170000                    MOV     #170000,R4      ;PUT MAINT REG DATA IN R4
006224  012702  177750                    MOV     #MAINT,R2       ;PUT ADDRESS OF MAINT ON R2
006230  005037  000370          9$:       CLR     @#370           ;ENSURE LOCATION 370 CLEAR
006234  012706  000376                    MOV     #376,SP         ;SETUP THE SP TO YELLOW ZONE
006240  000401                            BR      11$             ;GO TO 12$
        006242                            LOC=.                   ;THIS MAKES
        006240                            LOC=-3&LOC              ;THE NEXT INSTRUCTION
        006244                            LOC=LOC+4               ;FALL ON
        006244                            .-LOC                   ;AN EVEN WORD
006244  010412                  11$:      MOV     R4,(R2)         ;SET MAINT REG
006246  000240                            NOP                     ;ODD WORD GOOD PARITY
006250  005216                            INC     (SP)            ;CAUSE YEL ZONE (GOOD PARITY)
006252  005701                            TST     R1              ;ODD WORD BAD PARITY
                                ;SHOULD TAKE PE TRAP THEN YEL ZONE TRAP
006254  005012                  10$:      CLR     (R2)            ;CLEAR MAINTENANCE REGISTER
006256  000240                            NOP
006260  022737  000240  000370            CMP     #PR5,@#370      ;DID CACHVEC PSW GET STACKER?
006266  001403                            BEQ     12$             ;BRANCH IF YES
006270  012706  001400                    MOV     #STACK,SP       ;RESTORE THE SP
006274  104172                            ERROR   172             ;YEL ZONE CAME THRU ON PE TRAP
                                ;;**************************************************************
                                ;MEMORY MANAGEMENT TRAP DISABLED BY PARITY TRAP
006276  012737  006344  001510  12$:      MOV     #13$,$LPERR     ;SETUP ERROR LOOP
006304  012737  006374  000250            MOV     #15$,@#MMVEC    ;SETUP MEM MGT VECTOR
006312  012737  006374  000114            MOV     #15$,@#CACHVEC  ;SETUP CACHVEC
006320  012737  000340  000116            MOV     #PR7,@#CACHVEC+2 ;RESTORE EACH VEC PSW
006326  012704  170000                    MOV     #170000,R4      ;PUT MAINT DATA IN R4
006332  012702  177750                    MOV     #MAINT,R2       ;PUT ADDRESS OF MAINT REG IN R2
006336  112737  000004  172314            MOVB    #4,@#KIPDR6     ;ENSURE PAGE 6 TRAPS
006344  012706  001400          13$:      MOV     #STACK,SP       ;INITIALIZE THE SP
006350  012737  001001  177572            MOV     #1001,@#MMR0    ;TURN RELOCATION ON
006356  000402                            BR      16$
        006360                            LOC=.
        006360                            LOC=-3&LOC
        006364                            LOC=LOC+4
        006364                            .=LOC
```

```
006364  010412                          16$:    MOV     R4,(R2)         ;SET MAINT REG (PARITY GOOD)
006366  000240                                  NOP                     ;ODD WORD PARITY GOOD
006370  005237  140402                          INC     @#140402        ;INC HAS GOOD PARITY BUT ADDRESS
                                                                        ;HAS BAD PARITY. CAUSES MM TRAP
                                                                        ;AND PE TRAP
                                        ;TEST OK
006374  005012                          15$:    CLR     (R2)            ;CLEAR MAINT REG
006376  000240                                  NOP
006400  005037  177572                          CLR     @#MMR0          ;TURN RELOCATION OFF
006404  026627  000002  000340                  CMP     2(SP),#PR7      ;DID PE TRAP OCCUR FIRST?
006412  001401                                  BEQ     14$             ;BRANCH IF YES
006414  104173                                  ERROR   173             ;MEM MGT TRAP CAME
006416  012737  055412  000004          14$:    MOV     #CPSPUR,@#ERRVEC ;RESTORE LOCATION 4
006424  012737  055440  000114                  MOV     #SPUR,@#CACHVEC ;RESTORE LOCATION 114
006432  012737  177777  177744                  MOV     #-1,@#MEMERR    ;CLEAR MEM ERROR REG
006440  005037  177766                          CLR     @#CPUERR        ;ENSURE CPUERROR CLEAR
                                                                        ;CONTINUE
                                        ;
                                        ;;****************************************************************
                                        ;*      THE NEXT TEST USES THE MAPPING BOX AND THE CACHE TO
                                        ;*      GENERATE A PARITY ERROR ON THE UNIBUS.
                                        ;;****************************************************************
                                        ;*TEST 4        UNIBUS PARITY ERROR
                                        ;*
                                        ;*      THIS TEST MAKES A REFERENCE TO MEMORY THRU THE MAPPING
                                        ;*      BOX THAT WILL CAUSE A PARITY ERROR.  IF THE ABORT DOESN'T
                                        ;*      HAPPEN THEN THE PROBLEM IS ON UBCB.
                                        ;*
                                        ;*      NOTE:   MAP REGISTER 0 THRU 2 ARE NOT USED IN CASE THE PROGRAM
                                        ;*      IS RUNNING UNDER ACT11.
                                        ;;****************************************************************
006444  000004                          TST4:   SCOPE
006446  012737  077406  172314                  MOV     #77406,@#KIPDR6 ;SETUP PDR6
006454  012737  000060  172516                  MOV     #60,@#MMR3      ;SETUP MMR3
006462  012706  001400                          MOV     #STACK,SP       ;INITIALIZE THE SP
006466  012700  170220                          MOV     #MAPL4,R0       ;GET ADDRESS OF MAP REG 4
006472  012701  000032                          MOV     #32,R1          ;SETUP SOB COUNT
006476  012737  006510  000004                  MOV     #5$,@#ERRVEC    ;SETUP ERROR VECTOR
006504  005720                          8$:     TST     (R0)+           ;SEE IF MAP REG IS ENABLED
006506  000420                                  BR      6$              ;BRANCH IF YES
006510  062700  000002                  5$:     ADD     #2,R0           ;ADJUST R0 TO NEXT REGISTER
006514  077105                                  SOB     R1,8$           ;TEST NEXT REGISTER
006516  012706  001400                  7$:     MOV     #STACK,SP       ;RESTORE THE SP
006522  005737  001724                          TST     $PASS           ;FIRST PASS?
006526  001105                                  BNE     $EOT            ;BRANCH IF NO
006530  032777  040000  173002                  BIT     #SW14,@SWR      ;IS TEST BEING LOOPED ON?
006536  001101                                  BNE     $EOT            ;BRANCH IF YES
006540  104401  071730                          TYPE    ,EM724          ;TYPE MESSAGE
006544  000137  006742                          JMP     $EOT            ;GO TO NEXT TEST
006550  005010                          6$:     CLR     (R0)            ;ENSURE MAP REG HIGH CLEAR
006552  162700  000002                          SUB     #2,R0           ;GET ADDR OF MAP REG LOW
006556  012710  140000                          MOV     #140000,(R0)    ;PUT ADDR OF PAGE 6 IN MAP REG
006562  072027  000005                          ASH     #5,R0           ;ADJUST ADDR FOR PAR6
006566  052700  170000                          BIS     #170000,R0      ;SET UNIBUS ADDR BITS
006572  010037  172354                          MOV     R0,@#KIPAR6     ;PUT IN PAGE 6 PAR
006576  012737  005701  140000                  MOV     #5701,@#140000  ;PUT WORD WITH PAD PARITY IN 140000
```

```
006604  012704  170000                          MOV     #170000,R4      ;PUT MAINT REG DATA IN R4
006610  012702  177750                          MOV     #MAINT,R2       ;PUT ADDRESS OF MAINT REG IN R2
006614  012737  006636  001510                  MOV     #1$,$LPERR      ;SETUP ERROR LOOP
006622  012737  006700  000000                  MOV     #4$,@#0         ;SETUP LOCATION ZERO
006630  012737  006720  000114                  MOV     #2$,@#CACHVEC   ;SETUP CACH VECTOR
006636  012706  001400                  1$:     MOV     #STACK,SP       ;INITIALIZE THE SP
006642  052737  000001  177572                  BIS     #BIT0,@#MMR0    ;TURN RELOCATION ON
006650  000401                                  BR      3$              ;GO TO TEST
        006652                                  LOC=.
        006650                                  LOC=-3&LOC
        006654                                  LOC=LOC+4
        006654                                  .=LOC
006654  010412                          3$:     MOV     R4,(R2)         ;SET BITS IN MAINT REG
006656  000240                                  NOP                     ;GOOD PARITY ON ODD WORD
006660  005037  140000                          CLR     @#140000        ;EXECUTE A DATIP THRU THE
                                                                        ;MAP THAT CAUSES A PE
                                        ;FAILURE, NO ABORT
006664  005012                                  CLR     (R2)            ;CLEAR MAINT REG
006666  000240                                  NOP
006670  005037  177572                          CLR     @#MMR0          ;TURN RELOCATION OFF
006674  104174                                  ERROR   174             ;NO UNIBUS PE ABORT
006676  000410                                  BR      2$
                                        ;TRAPPED TO WRONG VECTOR
006700  005012                          4$:     CLR     (R2)            ;ENSURE MAINT REG CLEAR
006702  000240                                  NOP
006704  005037  177572                          CLR     @#MMR0          ;TURN OFF RELOCATION
006710  012737  177777  000004                  MOV     #-1,@#ERRVEC    ;CLEAR ERROR REGISTER
006716  104175                                  ERROR   175             ;TRAPPED TO ZERO
                                        ;TEST OK
006720  005012                          2$:     CLR     (R2)            ;ENSURE MAINT REG CLEAR
006722  000240                                  NOP
006724  005037  177572                          CLR     @#MMR0          ;TURN RELOCATION OFF
006730  012737  177777  177744                  MOV     #-1,@#MEMERR    ;CLEAR ERROR REG
006736  005037  172516                          CLR     @#MMR3          ;ENSURE MAP TURNED OFF
006742                                  $EOT:
                                        ;;*****************************************************************
                                        ;*TEST 5        CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FLOATING ONES
                                        ;*
                                        ;*THIS TEST IS A TEST OF BOTH THE AMX, CPU INPUTS, AND
                                        ;*THE CACHE ERROR ADDRESS REGISTER. A SET OF ADDRESSES IS
                                        ;*GENERATED AND A MAIN MEMORY ADDRESS AND CONTROL LINE
                                        ;*PARITY ERROR IS FORCED AT EACH, THEREBY LOCKING UP
                                        ;*THE ADDRESS ON THE OUTPUT OF  THE AMX IN THE ERROR
                                        ;*ADDRESS REGISTER. THE MANNER IN WHICH THIS IS DONE
                                        ;*IS AS FOLLOWS: FIRST THE ADDRESS IS GENERATED;
                                        ;*THEN, IF IT IS A VALID ADDRESS (THAT IS, IF IT IS NOT
                                        ;*BEYOND THE LIMITS OF MEMORY AS DISPLAYED IN THE
                                        ;*SYSTEM SIZE REGISTER), THESE THREE INSTRUCTIONS ARE MOVED
                                        ;*TO THAT AREA OF MEMORY:
                                        ;*      ONE:    MOV     R1,(R2)
                                        ;*      2$:     CLR     (R2)
                                        ;*      3$:     RTS     PC
                                        ;*2$ IS THE ADDRESS BEING TESTED. THE INSTRUCTION
                                        ;*AT ONE IS GIVEN CONTROL BY A 'JSR PC'. R1 IS MADE
                                        ;*TO CONTAIN #2 AND R2 CONTAINES THE ADDRESS OF
                                        ;*THE MAINTENANCE REGISTER, SO THAT AFTER THE 'MOV R1,(R2)'
                                        ;*IS EXECUTED A PARITY ERROR SHOULD OCCUR ON THE
                                        ;*MAIN MEMORY ADDRESS AND CONTROL LINES WHEN THE
                                        ;*NEXT INSTRUCTION IS FETCHED.
                                        ;*THE ADDRESSES USED ARE GENERATED FOLLOWINT THIS PATTERN
                                        ;*                200000
                                        ;*                200002
                                        ;*                200004
                                        ;*                2C0010
                                        ;*                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.
                                        ;*
                                        ;;**************************************************************
006742  000004                          TST5:   SCOPE
006744  012737  000020  001702                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000005                          X=$TN-1
                                                                        ;SET THE SKAD REGISTER
006752  012737  007650  055572                  MOV     #TST6,SKAD      ;IN CASE THE TEST ABORTS.

006760  113737  001502  001632                  MOVB    $TSTNM,$TMP0
006766  012737  055440  000114                  MOV     #SPUR,@#CACHVEC         ;INITIALLY EXPECT NO ERRORS

                                                                                ;SEE IF THIS TEST SHOULD
                                                                                ;BE EXECUTED. THE CONDITION
                                                                                ;TEST IS THE DESIGNATED
006774  104416                                  MMSKIP                          ;CONSOLE SWITCH.
006776  012700  172340                          MOV     #KIPAR0,R0              ;INITIALIZE THE KERNAL
007002  012701  077406                          MOV     #77406,R1               ;SPACE MEMORY MANAGEMENT
007006  012702  172300                          MOV     #KIPDR0,R2              ;REGISTERS
007012  012703  000010                          MOV     #10,R3
007016  010122                          1$:     MOV     R1,(R2)+
007020  077302                                  SOB     R3,1$
```

```
007022  005020                                  CLR     (R0)+
007024  012720  000200                          MOV     #200,(R0)+
007030  012720  000400                          MOV     #400,(R0)+
007034  012720  000600                          MOV     #600,(R0)+
007040  012720  001000                          MOV     #1000,(R0)+
007044  012720  001200                          MOV     #1200,(R0)+
007050  012720  001400                          MOV     #1400,(R0)+
007054  012710  177600                          MOV     #177600,(R0)
007060  012737  000020  172516                  MOV     #20,@#MMR3               ;TURN ON MEMORY MANAGEMENT
007066  012737  000001  177572                  MOV     #1,@#MMR0
007074  104417                                  SIZE                             ;DETERMINE FROM THE SYSTEM
                                                                                 ;SIZE REGISTER WHAT THE
                                                                                 ;HIGHEST ADDRESSABLE WORD
                                                                                 ;OF MEMORY IS.
007076  000000                          XLOADR: .WORD   0                        ;LOW ORDER 16-BITS OF THE
007100  000000                          XHIADR: .WORD   0                        ;ADDRESS AND HIGH ORDER 6-BITS
007102  042737  000002  007076                  BIC     #2,XLOADR                ;SET THE HIGHEST WORD MINUS TWO
                                                                                 ;IN XLOADR.

007110  012737  000014  177746                  MOV     #MOM1,@#CONTRL           ;FORCE MISSES TO BOTH GROUPS.

007116  005037  007636                          CLR     XADR3                    ;INITIALIZE STORAGE
007122  005037  007640                          CLR     XADR3+2                  ;LOCATIONS USED TO GENERATE
007126  005037  007626                          CLR     XADR1                    ;THE SERIES OF TEST ADDRESSES.
007132  012737  000001  007630                  MOV     #1,XADR1+2

007140                                  X1:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007140  023737  007630  007640                  CMP     XADR1+2,XADR3+2 ;COMPARE THE HIGH ORDER
007146  001006                                  BNE     64$             ;PARTS OF XADR1 AND ARG2.
007150  023737  007626  007636                  CMP     XADR1,XADR3     ;COMPARE THE LOW ORDER

007156  001002                                  BNE     64$             ;PARTS.



007160  000137  007602                          JMP     X11             ;THEY WERE EQUAL!

007164  103402                          64$:    BLO     65$
007166  000137  007176                          JMP     X2              ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND!
007172  000137  007602                  65$:    JMP     X11             ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.



007176                                  X2:
                                        ;DOUBLE PRECISION ADDITION, UNSIGNED
007176  013737  007626  007632                  MOV     XADR1,XADR2
007204  013737  007630  007634                  MOV     XADR1+2,XADR2+2
007212  063737  007636  007632                  ADD     XADR3,XADR2
007220  005537  007634                          ADC     XADR2+2
007224  063737  007640  007634                  ADD     XADR3+2,XADR2+2
```

```


007232                          X3:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007232  023737  007634  007100          CMP     XADR2+2,XLOADR+2        ;COMPARE THE HIGH ORDER
007240  001006                          BNE     64$             ;PARTS OF XADR2 AND ARG2.
007242  023737  007632  007076          CMP     XADR2,XLOADR    ;COMPARE THE LOW ORDER

007250  001002                          BNE     64$             ;PARTS.



007252  000137  007646                  JMP     XDONE           ;THEY WERE EQUAL!

007256  103402                  64$:    BLO     65$
007260  000137  007646                  JMP     XDONE           ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
007264  000137  007270          65$:    JMP     X4              ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.

007270  012737  007270  001510  X4:     MOV     #X4,$LPERR


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

007276  013703  007632                  MOV     XADR2,R3
007302  013702  007634                  MOV     XADR2+2,R2
007306  162703  000002                  SUB     #2,R3
007312  005602                          SBC     R2

007314  010300                          MOV     R3,R0
007316  042700  177701                  BIC     #177701,R0
007322  062700  140000                  ADD     #140000,R0
007326  073227  177772                  ASHC    #-6,R2
007332  010337  172354                  MOV     R3,@#KIPAR6

007336  012737  000020  172516          MOV     #20,@#MMR3      ;TURN ON MEMORY
007344  012737  000001  177572          MOV     #1,@#MMR0       ;MANAGEMENT.
                                                                ;SET UP THE TEST INSTRUCTIONS.
007352  012710  010112                  MOV     #010112,(R0)    ;010112 = 'MOV R1,(R2)'
007356  012760  005012  000002          MOV     #005012,2(R0)   ;005012 - 'CLR (R2)'
```

```
007364  012760  000207  000004          MOV     #000207,4(R0)       ;000207 = 'RTS PC'

007372  012701  000002                  MOV     #2,R1               ;SET UP THE REGISTERS
007376  012702  177750                  MOV     #MAINT,R2

007402  012737  007422  000114          MOV     #X6,@#CACHVEC       ;SET UP THE PARITY ERROR
007410  000240                          NOP                         ;TRAP VECTOR AND GO.
007412  004710                          JSR     PC,(R0)

007414                          X5:                                 ;NO TRAP OR ABORT OCCURRED.
                                                                              ;MAINTENANCE FUNCTION
                                                                              ;FOR BAD PARITY ON
007414  104022                  1$:     ERROR   22                            ;THE MAIN MEMORY ADDRESS
007416  000137  007534                  JMP     X9                            ;AND CONTROL LINES FAILED

007422                          X6:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
007422  023737  007634  177742          CMP     XADR2+2,LOADRS+2              ;COMPARE THE HIGH ORDER
007430  001006                          BNE     64$                 ;PARTS OF XADR2 AND ARG2.
007432  023737  007632  177740          CMP     XADR2,LOADRS        ;COMPARE THE LOW ORDER

007440  001002                          BNE     64$                 ;PARTS.




007442  000137  007460                  JMP     X7                  ;THEY WERE EQUAL!

007446  103402                  64$:    BLO     65$
007450  000137  007476                  JMP     X8                  ;THE FIRST ADDRESS IS LARGER
                                                                    ;THAN THE SECOND!
007454  000137  007476          65$:    JMP     X8                  ;THE FIRST IS LESS THAN THE
                                                                    ;SECOND.

                                                                              ;PARITY ERROR OCCURS.
007460  005726                  X7:     TST     (SP)+                         ;RESTORE THE STACK.
007462  022626                          CMP     (SP)+,(SP)+                   ;AND CONTINUE SINCE
007464  012737  177777  177744          MOV     #-1,@#MEMERR                  ;THE CACHE ERROR ADDRESS
007472  000137  007534                  JMP     X9                            ;REGISTER WAS SET CORRECTLY.

007476  013737  177744  001634  X8:     MOV     @#MEMERR,$TMP1                ;REPORT VALID TEST
                                                                              ;FAILURE.
007504  013737  177740  001640          MOV     @#LOADRS,$TMP3
007512  013737  177742  001642          MOV     @#HIADRS,$TMP4
007520  005726                          TST     (SP)+
007522  022626                          CMP     (SP)+,(SP)+
007524  104023                          ERROR   23
007526  012737  177777  177744          MOV     #-1,@#MEMERR

007534  005037  177572          X9:     CLR     @#MMR0                        ;TURN OFF MEMORY MANAGEMENT.
007540  005037  172516                  CLR     @#MMR3
007544  005737  007636                  TST     XADR3
007550  001007                          BNE     X10                           ;GET READY TO GENERATE
007552  005737  007640                  TST     XADR3+2                       ;THE NEXT TEST ADDRESS.
007556  001004                          BNE     X10
007560  012737  000002  007636          MOV     #2,XADR3
```

```
007566  000415                                  BR      X12

007570  006337  007636                  X10:    ASL     XADR3
007574  006137  007640                          ROL     XADR3+2
007600  000410                                  BR      X12

007602  006337  007626                  X11:    ASL     XADR1
007606  006137  007630                          ROL     XADR1+2
007612  005037  007636                          CLR     XADR3
007616  005037  007640                          CLR     XADR3+2
007622  000137  007140                  X12:    JMP     X1

007626  000000                          XADR1:  .WORD   0
007630  000000                                  .WORD   0
007632  000000                          XADR2:  .WORD   0
007634  000000                                  .WORD   0
007636  000000                          XADR3:  .WORD   0
007640  000000                                  .WORD   0
007642  000000                          XADR4:  .WORD   0
007644  000000                                  .WORD   0
007646  104414                          XDONE:  RSET                            ;DONE!

                                        ;;*****************************************************************
                                        ;*TEST 6         CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FLOATING ZEROES
                                        ;*
                                        ;*THIS IS ANOTHER TEST OF THE AMX WHICH IS CARRIED
                                        ;*OUT USING THE SAME METHOD AS IN THE PREVIOUS TEST
                                        ;*ALL THAT IS DIFFERENT IS THE SERIES OF TEST ADDRESSES
                                        ;*WHICH IS USED. IN THE PREVIOUS TEST A ONE WAS
                                        ;*FLOATED THROUGH A FIELD OF ZEROES TO PRODUCE THE
                                        ;*TEST ADDRESSES, HERE A ZERO WIL BE FLOATED THROUGH
                                        ;*A FIELD OF ONES TO PRODUCE THE ADDRESSES
                                        ;*BASE ADDRESSES WHICH ARE USE ARE:
                                        ;*              177776
                                        ;*              377776
                                        ;*              777776
                                        ;*              1777776
                                        ;*              3777776
                                        ;*              7777776
                                        ;*              17777776
                                        ;*EACH OF THESE PATTERNS IS TAKEN AND A ZERO IS FLOATED
                                        ;*THROUGHT THE FIELD OF ONES TO PRODUCE A TEST ADDRESS.
                                        ;*
                                        ;;*****************************************************************
007650  000004                          TST6:   SCOPE
007652  012737  000020  001702                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000006                          XX=$TN-1
                                                                        ;SET THE SKAD REGISTER
007660  012737  010550  055572                  MOV     #TST7,SKAD      ;IN CASE THE TEST ABORTS.

007666  113737  001502  001632                  MOVB    $TSTNM,$TMPO
007674  012737  055440  000114                  MOV     #SPUR,@#CACHVEC         ;INITIALLY EXPECT NO ERRORS.

007702  104416                                  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.
007704  012700  172340                  MOV     #KIPAR0,R0          ;INITIALIZE THE KERNAL MODE
007710  012701  077406                  MOV     #77406,R1           ;MEMORY MANAGEMENT REGISTERS
007714  012702  172300                  MOV     #KIPDR0,R2
007720  012703  000010                  MOV     #10,R3
007724  010122                  1$:     MOV     R1,(R2)+
007726  077302                          SOB     R3,1$
007730  005020                          CLR     (R0)+
007732  012720  000200                  MOV     #200,(R0)+
007736  012720  000400                  MOV     #400,(R0)+
007742  012720  000600                  MOV     #600,(R0)+
007746  012720  001000                  MOV     #1000,(R0)+
007752  012720  001200                  MOV     #1200,(R0)+
007756  012720  001400                  MOV     #1400,(R0)+
007762  012710  177600                  MOV     #177600,(R0)
007766  012737  000020  172516          MOV     #20,@#MMR3          ;TRUN ON MEMORY MANAGEMENT
007774  012737  000001  177572          MOV     #1,@#MMR0
010002  104417                          SIZE                  ;GET THE LARGEST MEMORY
010004  000000                  XXLOA:  .WORD   0                   ;WORD ADDRESS INTO XXLOA
010006  000000                  XXHIA:  .WORD   0                   ;AND XXHIA.
010010  042737  000002  010004          BIC     #2,XXLOA            ;GET THE ADDRESS OF THE HIGHEST WORD
                                                                    ;WORD MINUS TWO.



010016  012737  000014  177746          MOV     #MOM1,@#CONTRL      ;FROM NOW ON FORCE MISSES
                                                                    ;TO BOTH GROUPS.

010024  012737  177776  010526  XX1:    MOV     #177776,XXADR1      ;INITIALIZE
010032  005037  010530                  CLR     XXADR1+2
010036  012704  000016                  MOV     #16,R4
010042  000410                          BR      XX3

010044  005204                  XX2:    INC     R4                  ;TURN ON THE NEXT BIT
010046  052737  000001  010526          BIS     #1,XXADR1           ;IN THE FIELD OF ONES.
010054  006337  010526                  ASL     XXADR1
010060  006137  010530                  ROL     XXADR1+2

010064  012737  000002  010536  XX3:    MOV     #2,XXMASK           ;INITIALIZE THE MASK
010072  005037  010540                  CLR     XXMASK+2            ;USED TO CREATE THE ZERO
                                                                    ;IN THE FIELD OF ONES.
010076  010405                          MOV     R4,R5
010100  012737  010106  001510          MOV     #XX4,$LPERR

010106  013737  010526  010532  XX4:    MOV     XXADR1,XXADR2       ;DETERMINE THIS TEST ADDRESS.
010114  013737  010530  010534          MOV     XXADR1+2,XXADR2+2
010122  043737  010536  010532          BIC     XXMASK,XXADR2
010130  043737  010540  010534          BIC     XXMASK+2,XXADR2+2


                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010136  023737  010534  010544          CMP     XXADR2+2,XXCNST+2       ;COMPARE THE HIGH ORDER
010144  001006                          BNE     64$             ;PARTS OF XXADR2 AND ARG2.
010146  023737  010532  010542          CMP     XXADR2,XXCNST   ;COMPARE THE LOW ORDER
```

```

010154  001002                                  BNE     64$             ;PARTS.



010156  000137  010174                          JMP     XX5             ;THEY WERE EQUAL!

010162  103402                          64$:    BLO     65$
010164  000137  010174                          JMP     XX5             ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND.
010170  000137  010464                  65$:    JMP     XX10            ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.


010174                                  XX5:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010174  023737  010534  010006                  CMP     XXADR2+2,XXLOA+2        ;COMPARE THE HIGH ORDER
010202  001006                                  BNE     64$             ;PARTS OF XXADR2 AND ARG2.
010204  023737  010532  010004                  CMP     XXADR2,XXLOA    ;COMPARE THE LOW ORDER

010212  001002                                  BNE     64$             ;PARTS.



010214  000137  010232                          JMP     XX6             ;THEY WERE EQUAL!

010220  103402                          64$:    BLO     65$
010222  000137  010464                          JMP     XX10            ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND!
010226  000137  010232                  65$:    JMP     XX6             ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.


010232                                  XX6:


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

010232  013703  010532                          MOV     XXADR2,R3
010236  013702  010534                          MOV     XXADR2+2,R2
010242  162703  000002                          SUB     #2,R3
```

```
010246  005602                                  SBC     R2

010250  010300                                  MOV     R3,R0
010252  042700  177701                          BIC     #177701,R0
010256  062700  140000                          ADD     #140000,R0
010262  073227  177772                          ASHC    #-6,R2
010266  010337  172354                          MOV     R3,@#KIPAR6

010272  012737  000020  172516                  MOV     #20,@#MMR3          ;TURN ON MEMORY
010300  012737  000001  177572                  MOV     #1,@#MMR0           ;MANAGEMENT.
                                                                            ;SET UP THE TEST INSTRUCTIONS.
010306  012710  010112                          MOV     #010112,(R0)        ;010112 = 'MOV R1,(R2)'
010312  012760  005012  000002                  MOV     #005012,2(R0)       ;005012 = 'CLR (R2)'
010320  012760  000207  000004                  MOV     #000207,4(R0)       ;000207 = 'RTS PC'

010326  012701  000002                          MOV     #2,R1               ;SET UP THE REGISTERS
010332  012702  177750                          MOV     #MAINT,R2

010336  012737  010354  000114                  MOV     #XX7,@#CACHVEC      ;SET UP THE PARITY ERROR
010344  000240                                  NOP                         ;TRAP VECTOR AND GO.
010346  004710                                  JSR     PC,(R0)

                                                                            ;NO TRAP OCCURRED!
010350  104024                          1$:     ERROR   24
010352  000444                                  BR      XX10
                                        ;COME HERE ON THE PARITY ERROR
010354                                  XX7:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010354  023737  010534  177742                  CMP     XXADR2+2,LOADRS+2           ;COMPARE THE HIGH ORDER
010362  001006                                  BNE     64$                 ;PARTS OF XXADR2 AND ARG2.
010364  023737  010532  177740                  CMP     XXADR2,LOADRS       ;COMPARE THE LOW ORDER

010372  001002                                  BNE     64$                 ;PARTS.



010374  000137  010412                          JMP     XX8                 ;THEY WERE EQUAL!

010400  103402                          64$:    BLO     65$
010402  000137  010426                          JMP     XX9                 ;THE FIRST ADDRESS IS LARGER
                                                                            ;THAN THE SECOND!
010406  000137  010426                  65$:    JMP     XX9                 ;THE FIRST IS LESS THAN THE
                                                                            ;SECOND.


010412  005726                          XX8:    TST     (SP)+                       ;RESTORE THE STACK.
010414  022626                                  CMP     (SP)+,(SP)+
010416  012737  177777  177744                  MOV     #-1,@#MEMERR                ;RESET THE CACHE ERROR REGISTERS.
010424  000417                                  BR      XX10
010426  013737  177744  001634          XX9:    MOV     @#MEMERR,$TMP1              ;REPORT A VALID TEST
                                                                            ;FAILURE.
010434  013737  177740  001640                  MOV     @#LOADRS,$TMP3
010442  013737  177742  001642                  MOV     @#HIADRS,$TMP4
010450  005726                                  TST     (SP)+
010452  022626                                  CMP     (SP)+,(SP)+
010454  104025                            ERROR   25
010456  012737  177777  177744            MOV     #-1,@#MEMERR

010464  006337  010536           XX10:    ASL     XXMASK                          ;ROTATE THE MASK.
010470  006137  010540                    ROL     XXMASK+2
010474  005305                            DEC     R5
010476  001402                            BEQ     1$
010500  000137  010106                    JMP     XX4
010504  005037  177572           1$:      CLR     @#MMR0                          ;TURN OF MEMORY MANAGEMENT.
010510  005037  172516                    CLR     @#MMR3
010514  020427  000025                    CMP     R4,#25
010520  002012                            BGE     XX11
010522  000137  010044                    JMP     XX2

010526  000000                   XXADR1:  .WORD   0                               ;USED TO GENERATE TEST PATTERNS.
010530  000000                            .WORD   0
010532  000000                   XXADR2:  .WORD   0                               ;USED TO STORE THE CURRENT
010534  000000                            .WORD   0                               ;TEST PATTERN DURING A TEST.
010536  000000                   XXMASK:  .WORD   0                               ;MASK USED TO PUT A ZERO
010540  000000                            .WORD   0                               ;IN THE FIELD OF ONES
                                                                                  ;TO CREATE A TEST ADDRESS.
010542  126310                   XXCNST:  .WORD   BOTPRG                          ;THE SMALLEST ADDRESS
010544  000000                            .WORD   0                               ;IN MEMORY OVER THIS TEST.

010546  104414                   XX11:    RSET

                                 ;;*****************************************************************
                                 ;*TEST 7          CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ONES
                                 ;*
                                 ;*THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX.
                                 ;*THIS TEST IS IDENTICAL TO TST5 IN EVERY THING
                                 ;*IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE
                                 ;*REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM
                                 ;*THE CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL
                                 ;*GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS
                                 ;*WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE
                                 ;*AS UNIBUS REFERENCES.
                                 ;*
                                 ;;*****************************************************************
010550  000004                   TST7:    SCOPE
010552  012737  000020  001702            MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000007                   RR=$TN-1
                                                                  ;SET THE SKAD REGISTER
010560  012737  011462  055572            MOV     #TST10,SKAD     ;IN CASE THE TEST ABORTS.

010566  113737  001502  001632            MOVB    $TSTNM,$TMPO
010574  012737  055440  000114            MOV     #SPUR,@#CACHVEC         ;INITIALLY EXPECT NO ERRORS.
010602  012737  055412  000004            MOV     #CPSPUR,@#ERRVEC

010610  104416                            MMSKIP

010612  012700  172340                    MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
010616  012701  077406                    MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
010622  012702  172300                    MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
010626  012703  000010                    MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
010632  010122                   64$:     MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
```

```
010634  077302                                    SOB     R3,64$            ;ADDRESSES.
010636  005020                                    CLR     (R0)+
010640  012720  000200                            MOV     #200,(R0)+
010644  012720  000400                            MOV     #400,(R0)+
010650  012720  000600                            MOV     #600,(R0)+
010654  012720  001000                            MOV     #1000,(R0)+
010660  012720  001200                            MOV     #1200,(R0)+
010664  012720  001400                            MOV     #1400,(R0)+
010670  012710  177600                            MOV     #177600,(R0)

010674  012737  000060  172516                    MOV     #60,@#MMR3                  ;TURN ON MEMORY MANAGEMENT.
010702  012737  000001  177572                    MOV     #1,@#MMR0
                                                                                      ;DETERMINE THE MEMORY
010710  104417                                    SIZE                                ;SYSTEM SIZE.
010712  000000                          RRLOAD:   .WORD   0                           ;LOW ORDER 16-BITS AND
010714  000000                          RRHIAD:   .WORD   0                           ;HIGH ORDER 6-BITS OF THE
                                                                                      ;HIGHEST MEMORY WORD ADDRESS.
010716  042737  000002  010712                    BIC     #2,RRLOAD                   ;GET THE HIGHEST WORD IN MEMORY
                                                                                      ;MINUS TWO.
010724  012737  000014  177746                    MOV     #MOM1,@#CONTRL              ;FORCE MISSES TO BOTH GROUPS

010732  005037  011454                            CLR     RRADR3                      ;INITIALIZE STORAGE LOCATIONS
010736  005037  011456                            CLR     RRADR3+2                    ;USED TO GENERATE THE
010742  005037  011444                            CLR     RRADR1            ;SERIES OF TEST ADDRESSES.
010746  012737  000001  011446                    MOV     #1,RRADR1+2

010754                                  RR1:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
010754  023737  011446  011456                    CMP     RRADR1+2,RRADR3+2     ;COMPARE THE HIGH ORDER
010762  001006                                    BNE     64$               ;PARTS OF RRADR1 AND ARG2.
010764  023737  011444  011454                    CMP     RRADR1,RRADR3     ;COMPARE THE LOW ORDER

010772  001002                                    BNE     64$               ;PARTS.



010774  000137  011420                            JMP     RR11              ;THEY WERE EQUAL!

011000  103402                          64$:      BLO     65$
011002  000137  011012                            JMP     RR2               ;THE FIRST ADDRESS IS LARGER
                                                                            ;THAN THE SECOND!
011006  000137  011420                  65$:      JMP     RR11              ;THE FIRST IS LESS THAN THE
                                                                            ;SECOND.



011012                                  RR2:
                                        ;DOUBLE PRECISION ADDITION, UNSIGNED
011012  013737  011444  011450                    MOV     RRADR1,RRADR2
011020  013737  011446  011452                    MOV     RRADR1+2,RRADR2+2
011026  063737  011454  011450                    ADD     RRADR3,RRADR2
011034  005537  011452                            ADC     RRADR2+2
011040  063737  011456  011452                    ADD     RRADR3+2,RRADR2+2



```

```

011046                          RR3:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
011046  023737  011452  010714          CMP     RRADR2+2,RRLOAD+2       ;COMPARE THE HIGH ORDER
011054  001006                          BNE     64$             ;PARTS OF RRADR2 AND ARG2.
011056  023737  011450  010712          CMP     RRADR2,RRLOAD   ;COMPARE THE LOW ORDER

011064  001002                          BNE     64$             ;PARTS



011066  000137  011460                  JMP     RRDONE          ;THEY WERE EQUAL!

011072  103402                  64$:    BLO     65$
011074  000137  011460                  JMP     RRDONE          ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
011100  000137  011104          65$:    JMP     RR4             ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.

011104  012737  011104  001510  RR4:    MOV     #RR4,$LPERR
                                ;CONVERT THE PHYSICAL 22-BIT, ADDRESS IN RRADR2 TO A VIRTUAL ADDRESS
                                ;WHICH WILL RELOCATE THROUGH KIPAR6 TO THE UNIBUS, THEN THROUGH
                                ;THE MAPPING BOX TO THE UNIBUS INPUTS OF THE CACHE AMX.
                                ;NOTE: MAP REGISTERS 0-2 ARE NOT USED IN CASE PROGRAM IS
                                ;RUNNING UNDER APT OR ACT.
011112  013737  011450  170214          MOV     RRADR2,@#MAPL03 ;SET UP THE MAP REGISTER 3.
011120  013737  011452  170216          MOV     RRADR2+2,@#MAPH03
011126  162737  000002  170214          SUB     #2,@#MAPL03
011134  005637  170216                  SBC     @#MAPH03

011140  012700  140000                  MOV     #140000,R0      ;A VIRTUAL ADDRESS WHICH WILL
                                                                ;RELOCATE THROUGH KIPAR6.
011144  012737  170600  172354          MOV     #170600,@#KIPAR6;RELOCATE TO UNIBUS BASE
                                                                ;ADDRESS OF 000000.
011152  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAPPING BOX AND
                                                                ;22-BIT MODE.
011160  012737  000001  177572          MOV     #1,@#MMR0       ;TURN ON MEMORY MANAGEMENT.
                                                                ;SET UP THE TEST CODE:
011166  012710  010112                  MOV     #010112,(R0)    ;010112='MOV R1,(R2)'
011172  012760  005012  000002          MOV     #005012,2(R0)   ;005012='CLR (R2)'
011200  012760  000207  000004          MOV     #000207,4(R0)   ;000207='RTS PC'

011206  012701  000002                  MOV     #2,R1           ;SET UP THE REGISTERS USED
011212  012702  177750                  MOV     #MAINT,R2       ;IN THE TEST INSTRUCTIONS.

011216  012737  011236  000114          MOV     #RR6,@#CACHVEC  ;SET UP THE PARITY TRAP
011224  000240                          NOP                     ;VECTOR.
011226  004710                          JSR     PC,(R0) ;AND GO.


011230                          RR5:                            ;NO TRAP OR ABORT OCCURRED.
                                                                        ;MAINTENANCE FUNCTION FOR
011230  104030                  1$:     ERROR   30                      ;FORCING BAD PARITY ON
011232  000137  011352                  JMP     RR9                     ;THE MAIN MEMORY ADDRESS
                                                                        ;AND CONTROL LINES FAILED.
                                ;COME HERE WHEN THE FORCED ERROR OCCURS.
011236                          RR6:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
011236  023737  011452  177742            CMP     RRADR2+2,LOADRS+2       ;COMPARE THE HIGH ORDER
011244  001006                            BNE     64$             ;PARTS OF RRADR2 AND ARG2.
011246  023737  011450  177740            CMP     RRADR2,LOADRS   ;COMPARE THE LOW ORDER

011254  001002                            BNE     64$             ;PARTS.



011256  000137  011274                    JMP     RR7             ;THEY WERE EQUAL!

011'262 103402                  64$:      BLO     65$
011264  000137  011314                    JMP     RR8             ;THE FIRST ADDRESS IS LARGER
                                                                  ;THAN THE SECOND!
011270  000137  011314          65$:      JMP     RR8             ;THE FIRST IS LESS THAN THE
                                                                  ;SECOND.


011274  022626                  RR7:      CMP     (SP)+,(SP)+
011276  005726                            TST     (SP)+                   ;RESTORE THE STACK.
011300  022626                            CMP     (SP)+,(SP)+
011302  012737  177777  177744            MOV     #-1,@#MEMERR            ;CLEAR THE CACHE ERROR REGISTER.
011310  000137  011352                    JMP     RR9

011314  013737  177744  001634  RR8:      MOV     @#MEMERR,$TMP1          ;REPORT A VALID TEST FAILURE.
011322  013737  177740  001640            MOV     @#LOADRS,$TMP3
011330  013737  177742  001642            MOV     @#HIADRS,$TMP4
011336  005726                            TST     (SP)+
011340  022626                            CMP     (SP)+,(SP)+
011342  104031                            ERROR   31
011344  012737  000001  177744            MOV     #1,@#MEMERR             ;CLEAR THE ERROR REGISTER.
011352  005037  177572          RR9:      CLR     @#MMR0                  ;TURN OFF MEMORY MANAGEMENT.
011356  005037  172516                    CLR     @#MMR3
011362  005737  011454                    TST     RRADR3                  ;GET READY TO GENERATE THE
011366  001007                            BNE     RR10                    ;NEXT ADDRESS TO BE TESTED.
011370  005737  011454                    TST     RRADR3
011374  001004                            BNE     RR10
011376  012737  000002  011454            MOV     #2,RRADR3
011404  000415                            BR      RR12

011406  006337  011454          RR10:     ASL     RRADR3
011412  006137  011456                    ROL     RRADR3+2
011416  000410                            BR      RR12

011420  006337  011444          RR11:     ASL     RRADR1
011424  006137  011446                    ROL     RRADR1+2
011430  005037  011454                    CLR     RRADR3
011434  005037  011456                    CLR     RRADR3+2

011440  000137  010754          RR12:     JMP     RR1

011444  000000                  RRADR1:  .WORD    0                       ;3 DOUBLE WORD LOCATIONS
011446  000000                           .WORD    0                       ;USED TO STORE 22-BIT
```

```
011450  000000                          RRADR2: .WORD   0                                   ;ADDRESSES.
011452  000000                                  .WORD   0
011454  000000                          RRADR3: .WORD   0
011456  000000                                  .WORD   0

011460  104414                          RRDONE: RSET                                        ;DONE.

                                        ;;*****************************************************************
                                        ;*TEST 10        CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FLOATING ZEROES
                                        ;*
                                        ;*THIS IS A TEST OF THE UNIBUS INPUTS TO THE AMX.
                                        ;*THIS TEST IS IDENTICAL TO TST6 IN EVERY THING
                                        ;*IT DOES EXCEPT IN THAT TEST THE TEST ADDRESSES WERE
                                        ;*REFERENCED THROUGH MEMORY MANAGEMENT STRAIGHT FROM
                                        ;*THE CPU TO THE CACHE. HERE THE TEST ADDRESSES WILL
                                        ;*GO THROUGH THE MEMORY MANAGEMENT UNIT ONTO THE UNIBUS
                                        ;*WHERE THE MAPPING BOX WILL SEND THEM TO THE CACHE
                                        ;*AS UNIBUS REFERENCES.
                                        ;*
                                        ;;*****************************************************************
011462  000004                          TST10:  SCOPE
011464  012737  000020  001702                  MOV     #20,$TIMES          ;;DO 20 ITERATIONS
        000010                          SS=$TN-1
                                                                            ;SET THE SKAD REGISTER
011472  012737  012350  055572                  MOV     #TST11,SKAD         ;IN CASE THE TEST ABORTS.

011500  113737  001502  001632                  MOVB    $TSTNM,$TMP0
011506  012737  055440  000114                  MOV     #SPUR,@#CACHVEC             ;INITIALLY EXPECT NO ERRORS
011514  104416                                  MMSKIP

011516  012700  172340                          MOV     #KIPAR0,R0          ;INITIALLY PUT MEMORY
011522  012701  077406                          MOV     #77406,R1           ;MANAGEMENT IN A 'PASSIVE'
011526  012702  172300                          MOV     #KIPDR0,R2          ;STATE, THAT IS MAP ALL
011532  012703  000010                          MOV     #10,R3              ;VIRTUAL ADDRESSES ON TO
011536  010122                          64$:    MOV     R1,(R2)+            ;THEMSELVES AS PHYSICAL
011540  077302                                  SOB     R3,64$              ;ADDRESSES.
011542  005020                                  CLR     (R0)+
011544  012720  000200                          MOV     #200,(R0)+
011550  012720  000400                          MOV     #400,(R0)+
011554  012720  000600                          MOV     #600,(R0)+
011560  012720  001000                          MOV     #1000,(R0)+
011564  012720  001200                          MOV     #1200,(R0)+
011570  012720  001400                          MOV     #1400,(R0)+
011574  012710  177600                          MOV     #177600,(R0)

011600  104417                                  SIZE                                ;GET THE MEMORY SIZE.
011602  000000                          SSLOAD: .WORD   0                           ;22-BIT ADDRESS OF THE
011604  000000                          SSHIAD: .WORD   0                           ;HIGHEST WORD IN MEMORY.
011606  042737  000002  011602                  BIC     #2,SSLOAD                   ;GET THE HIGHEST WORD MINUS TWO.

011614  012737  000014  177746                  MOV     #MOM1,@#CONTRL

011622  012737  177776  012326  SS1:    MOV     #177776,SSADR1              ;INITIALIZE
011630  005037  012330                          CLR     SSADR1+2
011634  012704  000016                          MOV     #16,R4
011640  000410                                  BR      SS3
```

```

011642  005204                          SS2:    INC     R4                      ;TURN ON THE NEXT BIT
011644  052737  000001  012326                  BIS     #1,SSADR1               ;IN THE FIELD OF ONES
011652  006337  012326                          ASL     SSADR1
011656  006137  012330                          ROL     SSADR1+2

011662  012737  000002  012336          SS3:    MOV     #2,SSMASK               ;INITIALIZE THE MASK USER
011670  005037  012340                          CLR     SSMASK+2                ;TO CREATE THE ZERO IN
                                                                                ;IN FIELD OF ONES
011674  010405                                  MOV     R4,R5
011676  012737  011704  001510                  MOV     #SS4,$LPERR

011704  013737  012326  012332          SS4:    MOV     SSADR1,SSADR2           ;DETERMINE THE TEST ADDRESS.
011712  013737  012330  012334                  MOV     SSADR1+2,SSADR2+2
011720  043737  012336  012332                  BIC     SSMASK,SSADR2
011726  043737  012340  012334                  BIC     SSMASK+2,SSADR2+2

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
011734  023737  012334  012344                  CMP     SSADR2+2,SSCNST+2       ;COMPARE THE HIGH ORDER
011742  001006                                  BNE     64$             ;PARTS OF SSADR2 AND ARG2.
011744  023737  012332  012342                  CMP     SSADR2,SSCNST   ;COMPARE THE LOW ORDER

011752  001002                                  BNE     64$             ;PARTS.



011754  000137  011772                          JMP     SS5             ;THEY WERE EQUAL!

011760  103402                          64$:    BLO     65$
011762  000137  011772                          JMP     SS5             ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND!
011766  000137  012264                  65$:    JMP     SS10            ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.

011772                                  SS5:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
011772  023737  012334  011604                  CMP     SSADR2+2,SSLOAD+2       ;COMPARE THE HIGH ORDER
012000  001006                                  BNE     64$             ;PARTS OF SSADR2 AND ARG2.
012002  023737  012332  011602                  CMP     SSADR2,SSLOAD   ;COMPARE THE LOW ORDER

012010  001002                                  BNE     64$             ;PARTS.



012012  000137  012030                          JMP     SS6             ;THEY WERE EQUAL!

012016  103402                          64$:    BLO     65$
012020  000137  012264                          JMP     SS10            ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND!
012024  000137  012030                  65$:    JMP     SS6             ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.


012030                                  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.
                                ;NOTE: MAP REGISTERS 0-2 ARE NOT USED IN CASE PROGRAM IS
                                ;RUNNING UNDER APT OR ACT.
012030  013737  012332  170214          MOV     SSADR2,@#MAPL03 ;SET UP THE MAP REGISTER 3.
012036  013737  012334  170216          MOV     SSADR2+2,@#MAPH03
012044  162737  000002  170214          SUB     #2,@#MAPL03
012052  005637  170216                  SBC     @#MAPH03

012056  012700  140000                  MOV     #140000,R0         ;A VIRTUAL ADDRESS WHICH WILL
                                                                   ;RELOCATE THROUGH KIPAR6.
012062  012737  170600  172354          MOV     #170600,@#KIPAR6;RELOCATE TO UNIBUS BASE
                                                                   ;ADDRESS OF 000000.
012070  012737  000060  172516          MOV     #60,@#MMR3         ;TURN ON THE MAPPING BOX AND
                                                                   ;22-BIT MODE.
012076  012737  000001  177572          MOV     #1,@#MMR0          ;TURN ON MEMORY MANAGEMENT.
                                                                   ;SET UP THE TEST CODE:
012104  012710  010112                  MOV     #010112,(R0)       ;010112='MOV R1,(R2)'
012110  012760  005012  000002          MOV     #005012,2(R0)      ;005012='CLR (R2)'
012116  012760  000207  000004          MOV     #000207,4(R0)      ;000207='RTS PC'

012124  012701  000002                  MOV     #2,R1              ;SET UP THE REGISTERS USED
012130  012702  177750                  MOV     #MAINT,R2          ;IN THE TEST INSTRUCTIONS.

012134  012737  012210  000114          MOV     #SS8,@#CACHVEC     ;SET UP THE PARITY TRAP
012142  000240                          NOP                        ;VECTOR.
012144  004710                          JSR     PC,(R0) ;AND GO.

                                                                   ;NO TRAP OCCURRED!
012146  104032                  1$:     ERROR   32
012150  000445                          BR      SS10
                                ;TRAP TO HERE WHEN THE ERROR OCCURS.
012152                          SS7:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
012152  023737  012334  177742          CMP     SSADR2+2,LOADRS+2       ;COMPARE THE HIGH ORDER
012160  001006                          BNE     64$                ;PARTS OF SSADR2 AND ARG2.
012162  023737  012332  177740          CMP     SSADR2,LOADRS      ;COMPARE THE LOW ORDER

012170  001002                          BNE     64$                ;PARTS.



012172  000137  012210                  JMP     SS8                ;THEY WERE EQUAL!

012176  103402                  64$:    BLO     65$
012200  000137  012226                  JMP     SS9                ;THE FIRST ADDRESS IS LARGER
                                                                   ;THAN THE SECOND!
012204  000137  012226          65$:    JMP     SS9                ;THE FIRST IS LESS THAN THE
                                                                   ;SECOND.



012210  022626                  SS8:    CMP     (SP)+,(SP)+
012212  005726                          TST     (SP)+                          ;RESTORE THE STACK
012214  022626                          CMP     (SP)+,(SP)+
012216  012737  177777  177744          MOV     #-1,@#MEMERR                   ;CLEAR THE CACHE ERROR
```

```
012224  000417                                  BR      SS10                    ;REGISTER.

012226  013737  177744  001634  SS9:    MOV     @#MEMERR,$TMP1          ;REPORT A VALID TEST FAILURE.
012234  013737  177740  001640          MOV     @#LOADRS,$TMP3
012242  013737  177742  001642          MOV     @#HIADRS,$TMP4
012250  005726                          TST     (SP)+
012252  022626                          CMP     (SP)+,(SP)+
012254  104033                          ERROR   33
012256  012737  177777  177744          MOV     #-1,@#MEMERR

012264  006337  012336          SS10:   ASL     SSMASK                  ;ROTATE MASK TO FLOAT 0
012270  006137  012340                  ROL     SSMASK+2                ;TO THE LEFT.
012274  005305                          DEC     R5
012276  001402                          BEQ     1$
012300  000137  011704                  JMP     SS4
012304  005037  177572          1$:     CLR     @#MMR0                  ;TURN OF MEMORY MANAGEMENT
012310  005037  172516                  CLR     @#MMR3          ;AND THE MAPPING BOX.
012314  020427  000025                  CMP     R4,#25          ;IS THE TEST DONE?
012320  002012                          BGE     SS11            ;YES
012322  000137  011642                  JMP     SS2             ;NO

012326  000000                  SSADR1: .WORD   0               ;USED TO GENERATE THE
012330  000000                          .WORD   0               ;TEST ADDRESSESS.
012332  000000                  SSADR2: .WORD   0

012334  000000                          .WORD   0
012336  000000                  SSMASK: .WORD   0
012340  000000                          .WORD   0

012342  126310                  SSCNST: .WORD   BOTPRG          ;CONTAINS THE ADDRESS OF
012344  000000                          .WORD   0               ;THE LAST WORD OF THIS PROGRAM.

012346  104414                  SS11:   RSET                    ;DONE!

                                ;;*************************************************************
                                ;*TEST 11        CACHE ADDRESS MULTIPLEXER, AMX, CPU INPUTS DUAL ADDRESS TEST
                                ;*
                                ;*THIS TEST PERFORMS A DUAL ADDRESS TEST ON MEMORY LOCATED
                                ;*AT ADDRESSES LESS THAN 160000 (OCT.) OR WITHIN THE FIRST
                                ;*28K.  THE PURPOSE IS TO VARIFY THE THE AMX IS WORKING
                                ;*PROPERLY FOR THE LOW ORDER ADDRESS LINES INVOLVED.
                                ;*
                                ;;*************************************************************
012350  000004                  TST11:  SCOPE
012352  012737  000004  001702          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
        000011                  PP $TN-1
                                                                ;SET THE SKAD REGISTER
012360  012737  012606  055572          MOV     #TST12,SKAD     ;IN CASE THE TEST ABORTS.

012366  113737  001502  001632          MOVB    $TSTNM,$TMP0
012374  012737  055440  000114          MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.

012402  012737  000014  177746  PP1:    MOV     #M1M0,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS
012410  104417                          SIZE
012412  000000                  PPLOAD: .WORD   0               ;LOW ORDER 16-BITS AND
012414  000000                  PPHIAD: .WORD   0               ;HIGH ORDER 6-BITS OF THE
```

```
                                                                ;HIGHEST WORD ADDRESS IN
                                                                ;MEMORY.
012416  012737  157776  012602           MOV     #157776,PPLIM  ;ESTABLISH THE UPPER LIMIT
012424  005737  012414                   TST     PPHIAD         ;FOR THE TEST.
012430  001007                           BNE     PP2
012432  023737  012602  012412           LMP     PPLIM,PPLOAD
012440  003403                           BLE     PP2
012442  013737  012412  012602           MOV     PPLOAD,PPLIM

012450  012700  126310          PP2:     MOV     #BOTPRG,R0     ;THE LOW LIMIT FOR THIS TEST.
012454  010020                  1$:      MOV     R0,(R0)+       ;WRITE THE ADDRESS IN THE
012456  020037  012602                   CMP     R0,PPLIM       ;ADDRESS.
012462  101774                           BLOS    1$

012464  012700  126310                   MOV     #BOTPRG,R0
012470  011001                  PP3:     MOV     (R0),R1        ;GO BACK AND READ BACK THE
012472  020001                           CMP     R0,R1          ;ADDRESS, CHECK IT AND
012474  001411                           BEQ     PP4            ;WRITE BACK THE COMPLIMENT.
012476  010037  001644                   MOV     R0,$TMP5
                                                                ;REPORT ERROR.
012502  010137  001636                   MOV     R1,$TMP2
012506  010037  001640                   MOV     R0,$TMP3
012512  005037  001642                   CLR     $TMP4
012516  104034                  1$:      ERROR   34

012520  005120                  PP4:     COM     (R0)+          ;WRITE BACK COMPLIMENT.
012522  020037  012602                   CMP     R0,PPLIM
012526  101760                           BLOS    PP3

012530  012700  126310                   MOV     #BOTPRG,R0     ;GO BACK AND CHECK
012534  011001                  PP5:     MOV     (R0),R1        ;THE COMPLIMENTED PATTERNS.
012536  010002                           MOV     R0,R2
012540  005102                           COM     R2
012542  020102                           CMP     R1,R2
012544  001411                           BEQ     PP6
012546  010237  001644                   MOV     R2,$TMP5
012552  010137  001636                   MOV     R1,$TMP2
012556  010037  001640                   MOV     R0,$TMP3
012562  005037  001642                   CLR     $TMP4
012566  104034                  1$:      ERROR   34

012570  005120                  PP6:     COM     (R0)+
012572  020037  012602                   CMP     R0,PPLIM
012576  001356                           BNE     PP5
012600  000401                           BR      PP7

012602  000000                  PPLIM:   .WORD   0

012604  104414                  PP7:     RSET                   ;DONE!



                                ;;*******************************************************************
                                ;;*TEST 12      CACHE ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS DUAL ADDRESS TEST
                                ;;*
                                ;;*THIS TEST PERFORMS A DUAL ADDRESS TEST IDENTICAL TO
                                ;;*TST11, EXCEPT THAT IT IS DONE THROUGH THE MAPPING
```

```
                                          ;*BOX HERE THEREBY TESTING THE UNIBUS INPUTS TO THE AMX.
                                          ;*
                                          ;;************************************************************
012606  000004                            TST12:  SCOPE
012610  012737  000002  001702                    MOV     #2,$TIMES         ;;DO 2 ITERATIONS
        000012                            TT=$TN-1
                                                                            ;SET THE SKAD REGISTER
012616  012737  013246  055572                    MOV     #TST13,SKAD       ;IN CASE THE TEST ABORTS.

012624  113737  001502  001632                    MOVB    $TSTNM,$TMPO
012632  012737  055440  000114                    MOV     #SPUR,@#CACHVEC   ;EXPECT NO PARITY ERRORS.
012640  104416                                    MMSKIP
012642  012737  000014  177746            TT1:    MOV     #M1MO,@#CONTRL    ;FORCE MISSES TO BOTH GROUPS.
012650  104417                                    SIZE
012652  000000                            TTLOAD: .WORD   0                 ;DETERMINE THE HIGHEST
012654  000000                            TTHIAD: .WORD   0                 ;WORD IN MEMORY.

012656  012737  157776  013242                    MOV     #157776,TTLIM     ;DETERMINE THE UPPER LIMIT
012664  005737  012654                            TST     TTHIAD            ;FOR THE TEST.
012670  001007                                    BNE     TT2
012672  023737  013242  012652                    CMP     TTLIM,TTLOAD
012700  003403                                    BLE     TT2
012702  013737  012652  013242                    MOV     TTLOAD,TTLIM
012710                                    TT2:

012710  012700  172340                            MOV     #KIPAR0,R0        ;INITIALLY PUT MEMORY
012714  012701  077406                            MOV     #77406,R1         ;MANAGEMENT IN A 'PASSIVE'
012720  012702  172300                            MOV     #KIPDR0,R2        ;STATE, THAT IS MAP ALL
012724  012703  000010                            MOV     #10,R3            ;VIRTUAL ADDRESSES ON TO
012730  010122                            64$:    MOV     R1,(R2)+          ;THEMSELVES AS PHYSICAL
012732  077302                                    SOB     R3,64$            ;ADDRESSES.
012734  005020                                    CLR     (R0)+
012736  012720  000200                            MOV     #200,(R0)+
012742  012720  000400                            MOV     #400,(R0)+
012746  012720  000600                            MOV     #600,(R0)+
012752  012720  001000                            MOV     #1000,(R0)+
012756  012720  001200                            MOV     #1200,(R0)+
012762  012720  001400                            MOV     #1400,(R0)+
012766  012710  177600                            MOV     #177600,(R0)

012772  012737  000060  172516                    MOV     #60,@#MMR3        ;TURN ON MEMORY MANAGEMENT.
013000  012737  000001  177572                    MOV     #1,@#MMR0
013006  012700  126310                            MOV     #BOTPRG,R0        ;INITIALIZE A POINTER.

013012                                    1$:

013012  010037  170214                            MOV     R0,@#MAPL03       ;RELOCATE THE ADDRESS IN
013016  005037  170216                            CLR     @#MAPH03          ;R0 TO THE UNIBUS,
013022  012737  170600  172354                    MOV     #170600,@#KIPAR6;THROUGH THE MAPPING BOX
013030  012701  140000                            MOV     #140000,R1        ;TO THE CACHE.


013034  010011                                    MOV     R0,(R1)           ;WRITE THE ADDRESS IN THE
013036  062700  000002                            ADD     #2,R0             ;ADDRESS
013042  020037  013242                            CMP     R0,TTLIM
013046  101761                                    BLOS    1$
```

```

013050  012700  126310                  MOV     #BOTPRG,R0

013054                          TT3:

013054  010037  170214                  MOV     R0,@#MAPL03     ;RELOCATE THE ADDRESS IN
013060  005037  170216                  CLR     @#MAPH03        ;R0 TO THE UNIBUS,
013064  012737  170600  172354          MOV     #170600,@#KIPAR6;THROUGH THE MAPPING BOX
013072  012701  140000                  MOV     #140000,R1      ;TO THE CACHE.


013076  011102                          MOV     (R1),R2         ;READ BACK THE ADDRESS
013100  020002                          CMP     R0,R2           ;AS DATA IN THE LOCATION
013102  001411                          BEQ     TT4             ;IT ADDRESSES.
013104  010037  001644                  MOV     R0,$TMP5        ;REPORT ERROR IF NOT
                                                                ;EQUAL.
013110  010237  001636                  MOV     R2,$TMP2
013114  010037  001640                  MOV     R0,$TMP3
013120  005037  001642                  CLR     $TMP4
013124  104035                  1$:     ERROR   35
013126  005111                  TT4:    COM     (R1)            ;WRITE BACK THE
013130  062700  000002                  ADD     #2,R0           ;COMPLIMENTED DATA.
013134  020037  013242                  CMP     R0,TTLIM
013140  101745                          BLOS    TT3

013142  012700  126310                  MOV     #BOTPRG,R0

013146                          TT5:

013146  010037  170214                  MOV     R0,@#MAPL03     ;RELOCATE THE ADDRESS IN
013152  005037  170216                  CLR     @#MAPH03        ;R0 TO THE UNIBUS,
013156  012737  170600  172354          MOV     #170600,@#KIPAR6;THROUGH THE MAPPING BOX
013164  012701  140000                  MOV     #140000,R1      ;TO THE CACHE.


013170  011102                          MOV     (R1),R2         ;GO BACK AND CHECK
013172  010003                          MOV     R0,R3           ;THE COMPLIMENTED PATTERNS.
013174  005103                          COM     R3
013176  020203                          CMP     R2,R3
013200  001411                          BEQ     TT6
013202  010337  001644                  MOV     R3,$TMP5        ;REPORT ERROR
013206  010237  001636                  MOV     R2,$TMP2
013212  010037  001640                  MOV     R0,$TMP3
013216  005037  001642                  CLR     $TMP4
013222  104035                  1$:     ERROR   35

013224  005111                  TT6:    COM     (R1)            ;COMPLIMENT BACK THE DATA.
013226  062700  000002                  ADD     #2,R0
013232  020037  013242                  CMP     R0,TTLIM
013236  001343                          BNE     TT5
013240  000401                          BR      TT7

013242  000000                  TTLIM:  .WORD   0

013244  104414                  TT7:    RSET                    ;DONE!

```

```

                                                ;;*****************************************************************
                                                ;*TEST 13        CACHE ADDRESS MEMORY COMPARATOR TEST
                                                ;*
                                                ;*THIS IS A TEST OF THE CACHE ADDRESS MEMORY ADDRESS COMPARATORS.
                                                ;*THIS IS A CIRCUIT MADE UP OF SIX 74585 CHIPS, THREE FOR EACH
                                                ;*GROUP.  EACH CHIP COMPARES FOUR BITS OF THE ADDRESS ON THE
                                                ;*ADDRESS MULTIPLEXER, AMX, OUTPUT LINES WITH THE RESPECTIVE
                                                ;*FOUR BITS FROM THE CACHE ADDRESS MEMORY.  TWELVE BITS OF
                                                ;*THE ADDRESS ARE BROKEN DOWN THUS:  BITS 10 THROUGH 13
                                                ;*FOR THE FIRST COMPARATOR; BITS 14 THROUGH 17 FOR
                                                ;*THE NEXT; AND BITS 18 THROUGH 21 FOR THE LAST.
                                                ;*THE METHOD CHOSEN FOR THIS TEST IS TO TAKE EACH
                                                ;*POSSIBLE 4-BIT INPUT CONDITION FOR A COMPARATOR FROM THE
                                                ;*ADDRESS MEMORY AND PUT EVERY POSSIBLE 4-BIT COMBINATION
                                                ;*ON THE AMX SIDE OF THE COMPARATOR.  FOR 4-BITS
                                                ;*THERE ARE 16 (DEC) CONDITIONS.  THUS FOR EVERY 4-BIT
                                                ;*ADDRESS MEMORY INPUT TO THE COMPARATOR THERE ARE
                                                ;*16 AMX INPUT COMBINATIONS ONE OF WHICH WILL CAUSE
                                                ;*A MATCH AND MAKE THE REFERENCE A HIT.  THE OTHER
                                                ;*15 SHOULD OF COURSE BE MISSES.
                                                ;*
                                                ;;*****************************************************************
 013246  000004                                 TST13:  SCOPE
 013250  012737  000040  001702                         MOV     #40,$TIMES       ;;DO 40 ITERATIONS
                                                                                 ;SET THE SKAD REGISTER
 013256  012737  014420  055572                         MOV     #TST14,SKAD      ;IN CASE THE TEST ABORTS.

 013264  113737  001502  001632                         MOVB    $TSTNM,$TMPO
 013272  012737  055440  000114                         MOV     #SPUR,@#CACHVEC

 013300  104416                                         MMSKIP                   ;SEE IF THE SWITCH REGISTER
                                                                                 ;REFLECTS THE USERS DESIRE
                                                                                 ;TO ELIMINATE EXECUTION OF ANY TESTS
                                                                                 ;USING MEMORY MANAGEMENT.  IF
                                                                                 ;SO GO TO THE NEXT TEST.

 013302  012700  172340                                 MOV     #KIPAR0,R0       ;INITIALLY PUT MEMORY
 013306  012701  077406                                 MOV     #77406,R1        ;MANAGEMENT IN A 'PASSIVE'
 013312  012702  172300                                 MOV     #KIPDR0,R2       ;STATE, THAT IS MAP ALL
 013316  012703  000010                                 MOV     #10,R3           ;VIRTUAL ADDRESSES ON TO
 013322  010122                                 64$:    MOV     R1,(R2)+         ;THEMSELVES AS PHYSICAL
 013324  077302                                         SOB     R3,64$           ;ADDRESSES.
 013326  005020                                         CLR     (R0)+
 013330  012720  000200                                 MOV     #200,(R0)+
 013334  012720  000400                                 MOV     #400,(R0)+
 013340  012720  000600                                 MOV     #600,(R0)+
 013344  012720  001000                                 MOV     #1000,(R0)+
 013350  012720  001200                                 MOV     #1200,(R0)+
 013354  012720  001400                                 MOV     #1400,(R0)+
 013360  012710  177600                                 MOV     #177600,(R0)


 013364  104417                                         SIZE
 013366  000000                                 ZADLO:  .WORD   0                ;THE HIGHEST ADDRESSABLE
 013370  000000                                 ZADHI:  .WORD   0                ;MEMORY WORD AVAILABLE.
```

```

013372  005037  014164                  CLR     ZFLG1              ;ZFLG1 INDICATES WHICH GROUP
                                                                   ;IS BEING TESTED.
                                                                   ;ZFLG1 = 0, TESTING GROUP 0.
                                                                   ;ZFLG1 = 1, TESTING GROUP 1.
                                                                   ;TEST GROUP 0 FIRST.

013376  012737  000030  014172          MOV     #S0M1,ZGS          ;ZGS AND ZGM CONTAIN
013404  012737  000044  014170          MOV     #S1M0,ZGM          ;PATTERNS TO BE USED IN
                                                                   ;THE CACHE CONTROL REGISTER.
013412  005037  014166                  CLR     ZFLG2              ;ZFLG2 INDICATES WHICH
                                                                   ;4-BIT ADDRESS FIELD, OR
                                                                   ;WHICH COMPARATOR, IS
                                                                   ;BEING TESTED.
                                                                   ;ZFLG2 = 0, BITS 10 THROUGH 13
                                                                   ;ZFLG2 = 1, BITS 14 THROUGH 17
                                                                   ;ZFLG2 = 2, BITS 18 THROUGH 21
                                                                   ;ZFLG2 = 3, DONE!

013416  005737  014166          Z1:     TST     ZFLG2              ;SEE WHICH COMPARATOR
013422  001010                          BNE     Z2                 ;IS BEING TESTED ON THIS
                                                                   ;PASS AND PUT THE SIXTEEN
                                                                   ;POSS!BLE ADDRESSES NEEDED
                                                                   ;FOR THE TEST IN ZTABLE.
013424  012737  002000  014212          MOV     #2000,ZTABLE+4     ;BITS 10-13
013432  005037  014214                  CLR     ZTABLE+6
013436  004737  014310                  JSR     PC,ZCMTBL          ;CALL ZCMTBL TO FINISH THE TABLE.
013442  000432                          BR      Z5

013444  022737  000001  014166  Z2:     CMP     #1,ZFLG2
013452  001010                          BNE     Z3

013454  012737  040000  014212          MOV     #40000,ZTABLE+4    ;BITS 14-17
013462  005037  014214                  CLR     ZTABLE+6
013466  004737  014310                  JSR     PC,ZCMTBL          ;GET ZCMTBL TO FINISH SETTING
013472  000416                          BR      Z5                 ;UP THE TABLE.

013474  022737  000002  014166  Z3:     CMP     #2,ZFLG2
013502  001010                          BNE     Z4

013504  012737  000004  014214          MOV     #4,ZTABLE+6        ;BITS 18-21
013512  005037  014212                  CLR     ZTABLE+4
013516  004737  014310                  JSR     PC,ZCMTBL
013522  000402                          BR      Z5

013524  000137  014116          Z4:     JMP     Z14                ;DONE WITH THIS GROUP.

013530  012701  014176          Z5:     MOV     #ZTHR,R1
013534  013737  014170  177746          MOV     ZGM,@#CONTRL
013542  005711                          TST     (R1)               ;MAKE ZTHR A HIT IN BOTH GROUPS.
013544  013737  014172  177746          MOV     ZGS,@#CONTRL
013552  005711                          TST     (R1)

                                                                   ;FROM NOW ON SELECT THE GROUP BEING TESTED
                                                                   ;WHILE MISSING THE OTHER GROUP.
013554  012737  000020  172516          MOV     #20,@#MMR3         ;TURN ON MEMORY MANAGEMENT.
```

```
013562  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE!

013570  012701  014206                  MOV     #ZTABLE,R1      ;INITIALIZE R1 AS A POINTER
                                                                ;TO THE ADDRESS WHICH WILL
                                                                ;BE MADE A HIT.

013574                          Z7:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES


013574  023761  013370  000002          CMP     ZADLO+2,2(R1)   ;COMPARE THE HIGH ORDER
013602  001005                          BNE     64$             ;PARTS OF ZADLO AND (R1).
013604  023711  013366                  CMP     ZADLO,(R1)      ;THEN IF NECESSARY
013610  001002                          BNE     64$             ;COMPARE THE LOW ORDER PARTS.

013612  000137  013630                  JMP     1$              ;THEY WERE EQUAL!

013616  103402                  64$:    BLO     65$
013620  000137  013630                  JMP     1$              ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
013624  000137  014116          65$:    JMP     Z14             ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.


013630  012702  014206          1$:     MOV     #ZTABLE,R2      ;INITIALIZE A POINTER TO
                                                                ;THE ADDRESSES WHICH WILL
                                                                ;BE FED THROUGH THE COMPARATOR
                                                                ;AGAINST THE ADDRESS POINTED
                                                                ;TO BY THE OTHER POINTER, R1

013634  020102                  Z8:     CMP     R1,R2           ;DON'T TEST THE ADDRESS
013636  001511                          BEQ     Z12             ;AGAINST ITSELF HERE.

013640                          Z9:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES


013640  023762  013370  000002          CMP     ZADLO+2,2(R2)   ;COMPARE THE HIGH ORDER
013646  001005                          BNE     64$             ;PARTS OF ZADLO AND (R2).
013650  023712  013366                  CMP     ZADLO,(R2)      ;THEN IF NECESSARY
013654  001002                          BNE     64$             ;COMPARE THE LOW ORDER PARTS.

013656  000137  013674                  JMP     Z10             ;THEY WERE EQUAL!

013662  103402                  64$:    BLO     65$
013664  000137  013674                  JMP     Z10             ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
013670  000137  014074          65$:    JMP     Z13             ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.


013674                          Z10:

013674  011103                          MOV     (R1),R3         ;GET THE PHYSICAL ADDRESS POINTED
```

```
013676  042703  177700                   BIC     #177700,R3          ;TO BY R1 AND ESTABLISH
013702  011105                           MOV     (R1),R5 ;A VIRTUAL ADDRESS WHICH
013704  016104  000002                   MOV     2(R1),R4            ;WILL RELOCATE THROUGH
013710  073427  177772                   ASHC    #-6,R4              ;KIPAR6. SETUP KIPAR6 AND
013714  010537  172354                   MOV     R5,@#KIPAR6         ;LEAVE THE VIRTUAL ADDRESS
013720  062703  140000                   ADD     #140000,R3          ;IN R3.


013724  005713                           TST     (R3)
013726  005713                           TST     (R3)                ;SEE IF YOU CAN GET A HIT.
013730  032737  000010  177752           BIT     #10,@#HITMIS
013736  001011                           BNE     Z11
013740  013737  014164  001634           MOV     ZFLG1,$TMP1         ;NO! REPORT THE FAILURE
013746  011137  001636                   MOV     (R1),$TMP2
013752  016137  000002  001640           MOV     2(R1),$TMP3
013760  104026                   1$:     ERROR   26

013762                           Z11:

013762  011203                           MOV     (R2),R3             ;GET THE PHYSICAL ADDRESS POINTED
013764  042703  177700                   BIC     #177700,R3          ;TO BY R2 AND ESTABLISH
013770  011205                           MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
013772  016204  000002                   MOV     2(R2),R4            ;WILL RELOCATE THROUGH
013776  073427  177772                   ASHC    #-6,R4              ;KIPAR6. SETUP KIPAR6 AND
014002  010537  172354                   MOV     R5,@#KIPAR6         ;LEAVE THE VIRTUAL ADDRESS
014006  062703  140000                   ADD     #140000,R3          ;IN R3.


014012  000240                           NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
014014  005713                           TST     (R3)                ;MAKE SURE THERE IS NO
014016  032737  000010  177752           BIT     #10,@#HITMIS                ;MATCH.  A MISS?
014024  001416                           BEQ     Z12
014026  013737  014164  001634           MOV     ZFLG1,$TMP1         ;GOT A HIT! SO REPORT
014034  011137  001636                   MOV     (R1),$TMP2          ;FAILURE
014040  016137  000002  001640           MOV     2(R1),$TMP3
014046  011237  001642                   MOV     (R2),$TMP4
014052  016237  000002  001644           MOV     2(R2),$TMP5
014060  104027                   1$:     ERROR   27

014062  062702  000004           Z12:    ADD     #4,R2               ;MOVE POINTER TO NEXT AMX
                                                                     ;SIDE COMPARATOR INPUT ADDRESS.
014066  020227  014306                   CMP     R2,#ZTABOT          ;DONE?
014072  001260                           BNE     Z8                  ;BRANCH IF NOT DONE.

014074  062701  000004           Z13:    ADD     #4,R1               ;GO TO THE NEXT ADDRESS
014100  020127  014306                   CMP     R1,#ZTABOT          ;IN THE TABLE; OR IS THE
014104  001233                           BNE     Z7                  ;TEST USING THIS ADDRESS TABLE DONE?
                                                                     ;IF NOT GO TO Z7.
014106  005237  014166                   INC     ZFLG2               ;IF DONE WITH THESE ADDRESSES
014112  000137  013416                   JMP     Z1                  ;GO BACK TO COMPUTE THE
                                                                     ;NEXT ADDRESS TABLE, THAT IS
                                                                     ;CHECK THE NEXT 4-BIT
                                                                     ;COMPARATOR
014116  005037  177572           Z14:    CLR     @#MMR0              ;TURN OFF MEMORY MANAGEMENT.
014122  005037  172516                   CLR     @#MMR3
014126  005737  014164                   TST     ZFLG1               ;SEE IF BOTH GROUPS HAVE
```

```
014132 001131                          BNE     Z15             ;BEEN TESTED. BRANCH IF YES
014134 005237 014164                   INC     ZFLG1           ;OTHERWISE CHANGE THE
014140 012737 000044 014172            MOV     #S1M0,ZGS       ;PATTERNS USED IN THE CACHE
014146 012737 000030 014170            MOV     #S0M1,ZGM       ;CONTROL REGISTER AND GO
014154 005037 014166                   CLR     ZFLG2           ;BACK TO TEST GROUP 1.
014160 000137 013416                   JMP     Z1

014164 000000                  ZFLG1:  .WORD   0               ;FLAG WHICH DESIGNATES WHICH
                                                               ;GROUP IS BEING TESTED, 0 OR 1.
014166 000000                  ZFLG2:  .WORD   0               ;FLAG WHICH DESIGNATES WHICH
                                                               ;COMPARATOR IS BEING TESTED:
                                                               ;0 - BITS 10 THROUGH 13
                                                               ;1 - BITS 14 THROUGH 17
                                                               ;2 - BITS 18 THROUGH 21.

014170 000000                  ZGM:    .WORD   0               ;PATTERNS USED IN THE HIT
014172 000000                  ZGS:    .WORD   0               ;AND MISS REGISTER.
014174 000000                          .WORD   0
014176 000000                  ZTHR:   .WORD   0
014200 000000                          .WORD   0

014202 000000                  ZTMP1:  .WORD   0               ;TEMPORARY STORAGE LOCATIONS
014204 000000                  ZTMP2:  .WORD   0               ;USED BY THE ROUTINE, ZCMTBL,
                                                               ;TO GENERATE THE TEST ADDRESS
                                                               ;TABLE, ZTABLE.

014206 000040                  ZTABLE: .BLKW   40              ;THE TEST ADDRESS TABLE.
014306 000000                  ZTABOT: .WORD   0               ;PRECISION, 22-BIT, ADDRESSES.

                               ;THIS ROUTINE IS CALLED TO GENERATE THE TEST ADDRESS
                               ;TABLE, BY A 'JSR PC,ZCMTBL'.  IT CLEARS THE FIRST
                               ;ENTRY; IT ASSUMES THE THE BASE ADDRESS HAS BEEN
                               ;PLACED IN THE SECOND ENTRY BEFORE CONTROL IS PASSED
                               ;TO IT; THEN, STARTING WITH THE THIRD ENTRY, IT COMPUTES
                               ;EACH ENTRY BY ADDING THE BASE ADDRESS TO THE PRECEEDING
                               ;ENTRY.
014310 012701 014206           ZCMTBL: MOV     #ZTABLE,R1      ;ESTABLISH A POINTER TO
                                                               ;THE TABLE.
014314 005021                          CLR     (R1)+           ;CLR THE FIRST ENTRY.
014316 005021                          CLR     (R1)+
014320 012700 000016                   MOV     #16,R0
014324 012137 014202           1$:     MOV     (R1)+,ZTMP1     ;SAVE THE CURRENT ENTRY
014330 012137 014204                   MOV     (R1)+,ZTMP2
                                                               ;ADD THE OFFSET TO THE
                               ;DOUBLE PRECISION ADDITION, UNSIGNED



014334 013711 014202                   MOV     ZTMP1,(R1)
014340 013761 014204 000002            MOV     ZTMP1+2,2(R1)
014346 063711 014212                   ADD     ZTABLE+4,(R1)
014352 005561 000002                   ADC     2(R1)
014356 063761 014214 000002            ADD     ZTABLE+4+2,2(R1)
014364 077021                          SOB     R0,1$           ;LOOP UNTIL ZTABLE IS FILLED.
```

```
014366  012702  000020                          MOV     #20,R2
014372  012701  014206                          MOV     #ZTABLE,R1
014376  012700  014176                          MOV     #ZTHR,R0
014402  042700  176000                          BIC     #176000,R0
014406  060021                          2$:     ADD     R0,(R1)+
014410  005721                                  TST     (R1)+
014412  077203                                  SOB     R2,2$

014414  000207                                  RTS     PC                      ;THE RETURN

014416  104414                          Z15:    RSET                            ;DONE!


                                        ;;*******************************************************************
                                        ;*TEST 14        CACHE ADDRESS MEMORY COUNT PATTERN TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE ADDRESS MEMORY IN THE CACHE.
                                        ;*EVERY BIT IN THE MEMORY IS TURNED ON AND OFF WITHIN
                                        ;*THE LIMITATIONS OF MEMORY SIZE.  THE MANNER IN WHICH
                                        ;*THIS IS DONE IS TO ATTEMPT TO MAKE EVERY ADDRESS
                                        ;*IN AVAILABLE MEMORY A HIT IN EACH GROUP.
                                        ;*
                                        ;;*******************************************************************
014420  000004                          TST14:  SCOPE
014422  012737  000002  001702                  MOV     #2,$TIMES               ;;DO 2 ITERATIONS
        000014                          BB=$TN-1
014430                                  BB0:
                                                                                ;SET THE SKAD REGISTER
014430  012737  015450  055572                  MOV     #TST15,SKAD             ;IN CASE THE TEST ABORTS.

014436  113737  001502  001632                  MOVB    $TSTNM,$TMP0

014444  104416                                  MMSKIP

014446  104417                                  SIZE
014450  000000                          BBLOAD: .WORD   0
014452  000000                          BBHIAD: .WORD   0

014454  005037  015152                          CLR     BBFLG1                  ;TEST GROUP 0 FIRST.
014460  012737  000034  015162                  MOV     #S0M0M1,BBGS
014466  012737  000054  015164                  MOV     #S1M0M1,BBGM

014474  012737  055440  000114          BB1:    MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS, FOR NOW.
014502  012700  014430                          MOV     #BB0,R0                 ;MAKE THIS CODE HITS IN
014506  012701  001000                          MOV     #1000,R1                ;THE GROUP NOT BEING TESTED.
014512  013737  015162  177746          BB2:    MOV     BBGS,@#CONTRL
014520  005760  002000                          TST     2000(R0)
014524  013737  015164  177746                  MOV     BBGM,@#CONTRL
014532  005720                                  TST     (R0)+
014534  077112                                  SOB     R1,BB2

014536  013700  015162                          MOV     BBGS,R0                 ;FROM NOW ON FORCE
014542  042700  177717                          BIC     #177717,R0              ;SELECT THE GROUP BEING
014546  010037  177746                          MOV     R0,@#CONTRL             ;TESTED.

014552  012700  015136                  BB3:    MOV     #BBADR1,R0              ;INITIALIZE.
```

```
014556  012720  126310                   MOV     #BOTPRG,(R0)+     ;CONTAINS THE TEST ADDRESS.
014562  005020                           CLR     (R0)+
014564  005020                           CLR     (R0)+             ;CONTAINS THE LOGICAL 'OR'
014566  005020                           CLR     (R0)+             ;OF FAILING ADDRESSES.
014570  012720  177777                   MOV     #-1,(R0)+         ;CONTAINS THE LOGICAL 'AND'
014574  012720  177777                   MOV     #-1,(R0)+         ;OF BAD ADDRESSES


014600  012700  172340                   MOV     #KIPAR0,R0        ;INITIALLY PUT MEMORY
014604  012701  077406                   MOV     #77406,R1         ;MANAGEMENT IN A 'PASSIVE'
014610  012702  172300                   MOV     #KIPDR0,R2        ;STATE, THAT IS MAP ALL
014614  012703  000010                   MOV     #10,R3            ;VIRTUAL ADDRESSES ON TO
014620  010122                   64$:    MOV     R1,(R2)+          ;THEMSELVES AS PHYSICAL
014622  077302                           SOB     R3,64$            ;ADDRESSES.
014624  005020                           CLR     (R0)+
014626  012720  000200                   MOV     #200,(R0)+
014632  012720  000400                   MOV     #400,(R0)+
014636  012720  000600                   MOV     #600,(R0)+
014642  012720  001000                   MOV     #1000,(R0)+
014646  012720  001200                   MOV     #1200,(R0)+
014652  012720  001400                   MOV     #1400,(R0)+
014656  012710  177600                   MOV     #177600,(R0)

014662  012737  000020  172516           MOV     #20,@#MMR3        ;TURN ON MEMORY MANAGEMENT.
014670  012737  000001  177572           MOV     #1,@#MMR0

014676  005037  015154                   CLR     BBFLG2            ;INITIALIZE THE ERROR
014702  005037  015156                   CLR     BBCNT1            ;FLAG AND COUNT.
014706  005037  015160                   CLR     BBCNT1+2

014712  012737  015166  000114           MOV     #BBERR1,@#CACHVEC         ;PREPARF FOR ERRORS.

014720                           BB4:

                                 ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
014720  023737  014452  015140           CMP     BBLOAD+2,BBADR1+2         ;COMPARE THE HIGH ORDER
014726  001006                           BNE     64$               ;PARTS OF BBLOAD AND ARG2.
014730  023737  014450  015136           CMP     BBLOAD,BBADR1     ;COMPARE THE LOW ORDER

014736  001002                           BNE     64$               ;PARTS.



014740  000137  014756                   JMP     BB5               ;THEY WFRE EQUAL.

014744  103402                   64$:    BLO     65$
014746  000137  015054                   JMP     BB7               ;THE FIRST ADDRESS IS LARGER
                                                                   ;THAN THE SECOND!
014752  000137  014756           65$:    JMP     BB5               ;THE FIRST IS LESS THAN THE
                                                                   ;SECOND.


014756  012700  015136           BB5:    MOV     #BBADR1,R0        ;SET UP MEMORY MANAGEMENT.

014762  011003                           MOV     (R0),R3           ;GET THE PHYSICAL ADDRESS POINTED
014764  042703  177700                   BIC     #177700,R3        ;TO BY R0 AND ESTABLISH
```

```
014770  011005                             MOV     (R0),R5 ;A VIRTUAL ADDRESS WHICH
014772  016004  000002                     MOV     2(R0),R4            ;WILL RELOCATE THROUGH
014776  073427  177772                     ASHC    #-6,R4              ;KIPAR6. SETUP KIPAR6 AND
015002  010537  172354                     MOV     R5,@#KIPAR6         ;LEAVE THE VIRTUAL ADDRESS
015006  062703  140000                     ADD     #140000,R3          ;IN R3.


015012  000240                             NOP                         ;FOR SCOPING WITH AN OSCILLOSCOPE.
015014  005713                             TST     (R3)                ;TRY TO GET A HIT.
015016  005713                             TST     (R3)

015020  032737  000010  177752             BIT     #10,@#HITMIS        ;WAS IT A HIT?
015026  001004                             BNE     BB6                 ;BRANCH IF YES, OTHERWISE
                                                                       ;REPORT ERROR.
015030  013737  015152  001636             MOV     BBFLG1,$TMP2
015036  104036                    1$:      ERROR   36

015040  062737  000004  015136    BB6:     ADD     #4,BBADR1           ;MOVE TO NEXT WORD PAIR.
015046  005537  015140                     ADC     BBADR1+2
015052  000722                             BR      BB4

015054  005737  015154            BB7:     TST     BBFLG2              ;DID AN ERROR OCCUR IN
015060  001410                             BEQ     BB8                 ;THAT GROUP, IF YES PRINT
015062  112737  000037  001514             MOVB    #37,$ITEMB          ;AN ERROR SUMMARY
015070  013737  015152  001634             MOV     BBFLG1,$TMP1
015076  004737  056354                     JSR     PC,ERTYPE

015102  005737  015152            BB8:     TST     BBFLG1              ;HAVE BOTH GROUPS BEEN TESTED?
015106  001157                             BNE     BBDONE
015110  012737  000001  015152             MOV     #1,BBFLG1           ;IF NOT, GO BACK AND
015116  012737  000054  015162             MOV     #S1MOM1,BBGS        ;TEST GROUP 1
015124  012737  000034  015164             MOV     #S0MOM1,BBGM
015132  000137  014474                     JMP     BB1

015136  000000                    BBADR1:  .WORD   0                   ;THE TEST ADDRESS.
015140  000000                             .WORD   0
015142  000000                    BBADR2:  .WORD   0                   ;LOGICAL 'OR' OF BAD ADDRESSES.
015144  000000                             .WORD   0
015146  000000                    BBADR3:  .WORD   0                   ;LOGICAL 'AND' OF BAD ADDRESSES.
015150  000000                             .WORD   0

015152  000000                    BBFLG1:  .WORD   0                   ;FLAG: 1, IF TESTING GROUP 1,
                                                                       ;OR 0, IF TESTING GROUP 0.
015154  000000                    BBFLG2:  .WORD   0                   ;ERROR FLAG: 0, IF NO ERRORS
                                                                       ;OCCURRED IN THE TESTED
                                                                       ;GROUP.
015156  000000                    BBCNT1:  .WORD   0                   ;ERROR COUNT.
015160  000000                             .WORD   0

015162  000000                    BBGS:    .WORD   0                   ;PATTERNS FOR THE CACHE
015164  000000                    BBGM:    .WORD   0                   ;CONTROL REGISTER

015166                            BBERR1:

                                  ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
015166  023737  177742  015140             CMP     LOADRS+2,BBADR1+2        ;COMPARE THE HIGH ORDER
```

```
015174  001006                            BNE     64$               ;PARTS OF LOADRS AND ARG2.
015176  023737  177740  015136            CMP     LOADRS,BBADR1     ;COMPARE THE LOW ORDER

015204  001002                            BNE     64$               ;PARTS.



015206  000137  015224                    JMP     BBERR2            ;THEY WERE EQUAL!

015212  103402                    64$:    BLO     65$
015214  000137  055440                    JMP     SPUR              ;THE FIRST ADDRESS IS LARGER
                                                                    ;THAN THE SECOND.
015220  000137  055440            65$:    JMP     SPUR              ;THE FIRST IS LESS THAN THE
                                                                    ;SECOND.


015224  032737  000060  177744  BBERR2:   BIT     #60,@#MEMERR      ;MAKE SURE A CACHE ADDRESS
015232  001002                            BNE     BBERR3            ;MEMORY PARITY ERROR OCCURRED.
015234  000137  055440                    JMP     SPUR

015240                          BBERR3:                             ;REPORT ERROR.
015240  013737  015152  001640            MOV     BBFLG1,$TMP3
015246  012637  001636                    MOV     (SP)+,$TMP2
015252  005726                            TST     (SP)+
015254  013737  177744  001642            MOV     @#MEMERR,$TMP4
015262  013737  177740  001650            MOV     @#LOADRS,$TMP7
015270  013737  177742  001652            MOV     @#HIADRS,$TMP10
015276  013737  015136  001644            MOV     BBADR1,$TMP5
015304  013737  015140  001646            MOV     BBADR1+2,$TMP6
015312  104040                    1$:     ERROR   40

015314  053737  015136  015142            BIS     BBADR1,BBADR2      ;COMPUTE LOGICAL 'OR' OF
015322  053737  015140  015144            BIS     BBADR1+2,BBADR2+2         ;BAD ADDRESSES.
015330  005137  015146                    COM     BBADR3             ;COMPUT THE LOGICAL 'AND'
015334  043737  015136  015146            BIC     BBADR1,BBADR3      ;OF THE BAD ADDRESSES.
015342  005137  015146                    COM     BBADR3
015346  005137  015150                    COM     BBADR3+2
015352  043737  015140  015150            BIC     BBADR1+2,BBADR3+2
015360  005137  015150                    COM     BBADR3+2

015364  012737  177777  015154            MOV     #-1,BBFLG2         ;SET THE ERROR FLAG.
015372  005237  015156                    INC     BBCNT1             ;INCREMENT THE ERROR
015376  005537  015160                    ADC     BBCNT1+2           ;COUNT.

015402  012737  015424  000114            MOV     #BBERR4,@#CACHVEC         ;TRY TO GET THE BAD
                                                                     ;ADDRESS OUT OF THE ADDRESS
                                                                     ;MEMORY.
015410  013705  177740                    MOV     @#LOADRS,R5
015414  042705  176001                    BIC     #176001,R5
015420  005715                            TST     (R5)
015422  000401                            BR      BBERR5
015424  022626                  BBERR4:   CMP     (SP)+,(SP)+
015426  012737  177777  177744  BBERR5:   MOV     #-1,@#MEMERR
015434  012737  015166  000114            MOV     #BBERR1,@#CACHVEC
015442  000137  015040                    JMP     BB6

```

```
015446  104414                          BBDONE: RSET                    ;DONE!

                                        ;;*************************************************************
                                        ;*TEST 15        CACHE ADDRESS MEMORY PARITY LOGIC TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE PARITY CHECKERS AND PARITY GENERATOR
                                        ;*OF THE CACHE ADDRESS MEMORY. EVERY POSSIBLE ADDRESS TAG,
                                        ;*BITS 21 THROUGH 10, WHICH CAN BE STORED IN THE CACHE
                                        ;*ADDRESS MEMORY IS GENERATED, MADE A HIT AND THE
                                        ;*MAINTENANCE REGISTER IS THEN USED TO FORCE A CACHE ADDRESS
                                        ;*MEMORY PARITY ERROR AT EACH OF THE ADDRESSES
                                        ;*GENERATED. NOTE THAT BITS 9 THROUGH 0 OF THE ADDRESSES
                                        ;*IS NOT OF CONCERN, SO THESE BITS WILL BE THE SAME
                                        ;*FOR EACH ADDRESS; THIS IS BECAUSE ONLY BITS 21 THROUGH
                                        ;*10 ARE STORED IN THE ADDRESS MEMORY THEREFORE ONLY
                                        ;*THESE BITS ARE PARITY CHECKED IN THE CACHE ADDRESS
                                        ;*MEMORY PARITY CHECKERS. ALSO NOTE THAT THE RANGE
                                        ;*OF THE ADDRESSES MUST BE LIMITED TO BETWEEN THE
                                        ;*BOUNDS IMPOSED BY THE HIGHEST AVAILABLE MEMORY WORD
                                        ;*AND THE LAST WORD OF MEMORY USED BY THIS PROGRAM.
                                        ;*THE MANNER IN WHICH THE ERROR WILL BE FORCED
                                        ;*WILL BE TO PUT THE INSTRUCTIONS:
                                        ;*       1$:     MOV     R4,(R2)
                                        ;*       TSTADS: CLR     (R2)
                                        ;*               RTS     PC
                                        ;*AT THE PARTICULAR ADDRESS BEING TESTED, WHERE
                                        ;*'TSTADS' IS THE ADDRESS BEING TESTED. R4 CONTAINS
                                        ;*A PATTERN TO BE LOADED IN THE MAINTENANCE REGISTER
                                        ;*WHICH WILL FORCE AN ERROR IN THE CACHE ADDRESS
                                        ;*MEMORY; R2 CONTAINS THE ADDRESS OF THE MAINTENANCE
                                        ;*REGISTER. NOTE FOR EACH ADDRESS R4 WILL FIRST
                                        ;*BE SUCH AS TO CAUSE AN ERROR IN THE LOW
                                        ;*BYTE ADDRESS PARITY CHECKER THEN AT THE SAME
                                        ;*ADDRESS AN ERROR WILL BE FORCED ON THE HIGH BYTE.
                                        ;*THE SEQUENCE OF TEST ADDRESSES WILL BE GENERATED
                                        ;*TWICE ONCE MAKING THEM HITS IN GROUP 0 THEN
                                        ;*MAKING THEM HITS IN GROUP 1.
                                        ;*
                                        ;;*************************************************************
015450  000004                          TST15:  SCOPE
015452  012737  000002  001702                  MOV     #2,$TIMES       ;;DO 2 ITERATIONS
        000015                          AA=$TN-1
                                                                        ;SET THE SKAD REGISTER
015460  012737  016540  055572                  MOV     #TST16,SKAD     ;IN CASE THE TEST ABORTS.

015466  113737  001502  001632                  MOVB    $TSTNM,$TMPO
015474  012737  055440  000114                  MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
015502  104416                                  MMSKIP

015504  012700  172340                          MOV     #KIPARO,R0      ;INITIALLY PUT MEMORY
015510  012701  077406                          MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
015514  012702  172300                          MOV     #KIPDRO,R2      ;STATE, THAT IS MAP ALL
015520  012703  000010                          MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
015524  010122                          64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
015526  077302                                  SOB     R3,64$          ;ADDRESSES.
015530  005020                                  CLR     (R0)+
015532  012720  000200                    MOV     #200,(R0)+
015536  012720  000400                    MOV     #400,(R0)+
015542  012720  000600                    MOV     #600,(R0)+
015546  012720  001000                    MOV     #1000,(R0)+
015552  012720  001200                    MOV     #1200,(R0)+
015556  012720  001400                    MOV     #1400,(R0)+
015562  012710  177600                    MOV     #177600,(R0)

015566  104417                            SIZE
015570  000000                  AALOAD:   .WORD   0                ;ADDRESS OF THE HIGHEST
015572  000000                  AAHIAD:   .WORD   0                ;WORD IN MEMORY.
015574  042737  000002  015570            BIC     #2,AALOAD

015602  012700  016400                    MOV     #AATMP2,R0       ;ESTABLISH BITS 9 THROUGH
015606  042700  176003                    BIC     #176003,R0       ;0 TO BE PART OF ALL
015612  010037  016364                    MOV     R0,AAOFST        ;THE TEST ADDRESSES.
015616  005037  016366                    CLR     AAOFST+2

015622  012737  000020  172516            MOV     #20,@#MMR3       ;ENABLE 22-BIT MODE
015630  012737  000001  177572            MOV     #1,@#MMR0        ;ADDRESSING

015636  012737  000030  016354            MOV     #SOM1,AAGS       ;TEST GROUP 0 FIRST, AAGS
015644  005037  016350                    CLR     AAFLG1           ;CONTAINS A PATTERN TO
015650  012737  001400  016356            MOV     #1400,AAERGS     ;BE PUT IN THE CONTROL
015656  012737  004420  016374            MOV     #4420,AAEXER     ;REGISTER. AAERGS CONTAINS
                                                                   ;A PATTERN FOR THE MAINT. REG.
015664  012737  000001  016362  AA1:      MOV     #1,AAADR1+2      ;AAADR1 CONTAINS BITS
015672  005037  016360                    CLR     AAADR1           ;10 THROUGH 22 OF
                                                                   ;THE TEST ADDRESS.
                                                                   ;INITIALIZE IT.
015676  013737  016354  177746            MOV     AAGS,@#CONTRL    ;SELECT THE GROUP BEING
                                                                   ;TESTED. MISS THE OTHER
                                                                   ;GROUP.
015704                          AA2:                               ;GET THE TEST ADDRESS
                                                                   ;INTO THE AAADR2=AAADR1+AAOFST
                                ;DOUBLE PRECISION ADDITION, UNSIGNED
015704  013737  016360  016370            MOV     AAADR1,AAADR2
015712  013737  016362  016372            MOV     AAADR1+2,AAADR2+2
015720  063737  016364  016370            ADD     AAOFST,AAADR2
015726  005537  016372                    ADC     AAADR2+2
015732  063737  016366  016372            ADD     AAOFST+2,AAADR2+2



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

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
015740  023737  016372  015572            CMP     AAADR2+2,AALOAD+2     ;COMPARE THE HIGH ORDER
015746  001006                            BNE     64$              ;PARTS OF AAADR2 AND ARG2.
015750  023737  016370  015570            CMP     AAADR2,AALOAD    ;COMPARE THE LOW ORDER

015756  001002                            BNE     64$              ;PARTS.

```

```


015760  000137  015776                  JMP     AA3             ;THEY WERE EQUAL!

015764  103402                  64$:    BLO     65$
015766  000137  016306                  JMP     AA8             ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
015772  000137  015776          65$:    JMP     AA3             ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.


015776  012737  000001  016352  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.
016004  013703  016370                  MOV     AAADR2,R3
016010  013702  016372                  MOV     AAADR2+2,R2
016014  162703  000002                  SUB     #2,R3
016020  005602                          SBC     R2
016022  010300                          MOV     R3,R0
016024  042700  177700                  BIC     #177700,R0      ;R0 CONTAINS THE VIRTUAL
016030  062700  140000                  ADD     #140000,R0      ;ADDRESS.

016034  073227  177772                  ASHC    #-6,R2          ;SET KIPAR6
016040  010337  172354                  MOV     R3,@#KIPAR6

016044  012737  055440  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
016052  012710  010112                  MOV     #010112,(R0)    ;010112 = 'MOV R4,(R2)'
016056  012760  005012  000002          MOV     #005012,2(R0)   ;005012 = 'CLR (R2)'
016064  012760  000207  000004          MOV     #000207,4(R0)   ;000207 = 'RTS PC'

016072  005760  000002                  TST     2(R0)           ;MAKE THE TEST ADDRESS
016076  005760  000002                  TST     2(R0)           ;A HIT IN THE GROUP
016102  032737  000010  177752          BIT     #10,@#HITMIS    ;BEING TESTED!
016110  001016                          BNE     AA4

016112  012737  016140  001640          MOV     #1$,$TMP3       ;IF UNABLE TO GET A GIT
016120  013737  016350  001634          MOV     AAFLG1,$TMP1    ;REPORT ERROR!
016126  010037  001636                  MOV     R0,$TMP2
016132  062737  000002  001636          ADD     #2,$TMP2
016140  104001                  1$:     ERROR   1
016142  000137  016270                  JMP     AA7             ;GO TO NEXT TEST ADDRESS.

016146                          AA4:                            ;THE TEST ADDRESS IS NOW
                                                                ;A HIT IN THE GROUP
016146  012737  016404  000114          MOV     #AAERR1,@#CACHVEC       ;BEING TESTED. NOW RESET
                                                                ;CACHVEC TO GO TO THE EXPECTED
                                                                ;ERROR HANDLER
```

```
016154 012702 177750                   MOV    #MAINT,R2           ;SET R2
016160 013704 016356                   MOV    AAERGS,R4           ;SET R4 WHICH WILL BE
016164 042704 005000                   BIC    #5000,R4            ;LOADED INTO THE MAINT.
                                                                  ;REG SO AS TO FORCE
                                                                  ;A LOW BYTE ADDRESS
                                                                  ;MEMORY PARITY ERROR
                                                                  ;IN THE GROUP BEING
                                                                  ;TESTED.
016170 000240                          NOP                        ;FOR SCOPING WITH AN OSCILLOSCOPE.
016172 004710                          JSR    PC,(R0)             ;GO TO THE TEST
                                                                  ;ADDRESS!

016174                          AA5:                              ;RETURN,RTS PC, BACK TO
016174 013737 016350 001636            MOV    AAFLG1,$TMP2        ;HERE IF THE TEST FAILED
016202 013737 016370 001640            MOV    AAADR2,$TMP3        ;TO FORCE AN ERROR AT
016210 013737 016372 001642            MOV    AAADR2+2,$TMP4      ;THE TEST ADDRESS'S LOW
016216 104136                   1$:    ERROR  136                 ;BYTE. REPORT THE FAILURE!

016220                          AA6:                              ;TRY TO DO THE SAME
                                                                  ;THING NOW ONLY FORCE THE
                                                                  ;ERROR ON THE ADDRESSES
                                                                  ;HIGH BYTE!
                                                                  ;THE INSTRUCTIONS ARE
                                                                  ;ALREADY AT THE TEST
016220 012737 000002 016352            MOV    #2,AAFLG2           ;ADDRESS. BUT MAKE SURE
016226 005760 000002                   TST    2(R0)               ;IT IS STILL A HIT!
016232 013704 016356                   MOV    AAERGS,R4           ;SET R4 TO FORCE THE
016236 042704 002400                   BIC    #2400,R4            ;ERROR ON THE HIGH BYTE.
016242 004710                          JSR    PC,(R0)             ;GO DO THE TEST!

016244                          AA16:                             ;RETURN,RTS PC, BACK TO HERE
016244 013737 016350 001636            MOV    AAFLG1,$TMP2        ;IF THE TEST FAILED
016252 013737 016370 001640            MOV    AAADR2,$TMP3        ;IN TRYING TO FORCE A
016260 013737 016372 001642            MOV    AAADR2+2,$TMP4      ;ERROR ON THE HIGH BYTE
016266 104137                   1$:    ERROR  137                 ;IN THE ADDRESS MEMORY

016270 062737 002000 016360     AA7:   ADD    #2000,AAADR1        ;INCREMENT BITS 21 THROUGH
016276 005537 016362                   ADC    AAADR1+2            ;10 OF THE TEST ADDRESS
016302 000137 015704                   JMP    AA2                 ;AND GO TEST THIS NEW
                                                                  ;TEST ADDRESS!
016306 005737 016350            AA8:   TST    AAFLG1              ;SEE IF BOTH GROUPS HAVE
016312 001111                          BNE    AADONE              ;BEEN TESTED. IF NOT, GO
016314 012737 004440 016374            MOV    #4440,AAEXER        ;BACK TO AA1 TO TEST
016322 012737 000044 016354            MOV    #S1M0,AAGS          ;GROUP ONE, OTHERWISE DONE!
016330 012737 000001 016350            MOV    #1,AAFLG1
016336 012737 006000 016356            MOV    #6000,AAERGS
016344 000137 015664                   JMP    AA1

016350 000000                   AAFLG1: .WORD 0                   ;A FLAG WHICH INDICATES
                                                                  ;WHICH GROUP IS BEING TESTED
                                                                  ;1 OR 0
016352 000000                   AAFLG2: .WORD 0                   ;A FLAG WHICH INDICATES
                                                                  ;WHETHER THE LOW BYTE (1)
                                                                  ;THE HIGH BYTE (2) IS
                                                                  ;BEING TESTED.
016354 000000                   AAGS:   .WORD 0                   ;A PATTERN FOR THE CONTROL
```

```
                                                                       ;REGISTER.
016356  000000                          AAERGS: .WORD   0              ;PATTERN FOR THE MAINT. REG.
016360  000000                          AAADR1: .WORD   0              ;BITS 21 THROUGH 10 OF
016362  000000                                  .WORD   0              ;THE TEST ADDRESS.
016364  000000                          AAOFST: .WORD   0              ;BITS 9 THROUGH 0 OF
016366  000000                                  .WORD   0              ;THE TEST ADDRESS.
016370  000000                          AAADR2: .WORD   0              ;THE TEST ADDRESS
016372  000000                                  .WORD   0              ;'AAADR2 = AAADR1+AAOFST'
016374  000000                          AAEXER: .WORD   0              ;EXPECTED ERROR REGISTER
016376  000000                          AATMP1: .WORD   0              ;THESE ADDRESSES ARE
016400  000000                          AATMP2: .WORD   0              ;USED TO DETERMINE AAOFST.
016402  000000                                  .WORD   0

016404  016666  000002  000004          AAERR1: MOV     2(SP),4(SP)    ;RESET THE STACK. RECALL THAT THE
016412  012616                                  MOV     (SP)+,(SP)               ;TEST ROUTINE WAS JSR'ED TO AND
                                                                       ;A PARITY ERROR TRAP BROUGHT CONTROL
                                                                       ;BACK!!
016414  023737  016374  177744                  CMP     AAEXER,@#MEMERR ;MAKE SURE THE ERROR
016422  001405                                  BEQ     1$             ;WHICH OCCURRED WAS
016424  012737  055440  000114                  MOV     #SPUR,@#CACHVEC ;THE EXPECTED ERROR AT
016432  000137  055440                          JMP     SPUR           ;THE EXPECTED ADDRESS,
                                                                       ;IF NOT GO TO THE
                                                                       ;SPURIOUS ERROR HANDLER,
                                                                       ;SPUR!
016436                                  1$:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
016436  023737  016372  177742                  CMP     AAADR2+2,LOADRS+2        ;COMPARE THE HIGH ORDER
016444  001006                                  BNE     64$            ;PARTS OF AAADR2 AND ARG2.
016446  023737  016370  177740                  CMP     AAADR2,LOADRS  ;COMPARE THE LOW ORDER

016454  001002                                  BNE     64$            ;PARTS.



016456  000137  016474                          JMP     AAERR2         ;THEY WERE EQUAL!

016462  103402                          64$:    BLO     65$
016464  000137  055440                          JMP     SPUR           ;THE FIRST ADDRESS IS LARGER
                                                                       ;THAN THE SECOND!
016470  000137  055440                  65$:    JMP     SPUR           ;THE FIRST IS LESS THAN THE
                                                                       ;SECOND.


016474  012737  177777  177744          AAERR2: MOV     #-1,@#MEMERR   ;IF EVERYTHING WAS
                                                                                 ;CORRECT, CLR THE ERROR
016502  022626                                  CMP     (SP)+,(SP)+    ;REGISTER RESET THE
                                                                       ;STACK AND CONTINUE
016504  023727  016352  000002                  CMP     AAFLG2,#2      ;TESTING
016512  001002                                  BNE     1$
016514  000137  016270                          JMP     AA7            ;TEST THE NEXT ADDRESS
016520  023727  016352  000001          1$:     CMP     AAFLG2,#1
016526  001002                                  BNE     2$
016530  000137  016220                          JMP     AA6            ;TEST THE HIGH BYTE OF THIS ADDRESS
016534  000000                          2$:     HALT                   ;???HOW DID WE GET HERE?
```

```
016536  104414                          AADONE: RSET                            ;DONE'



                                        ;;*************************************************************
                                        ;*TEST 16       CACHE ADDRESS MEMORY DUAL ADDRESS TEST, UPWARD
                                        ;*
                                        ;*THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS
                                        ;*MEMORY.  AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS'
                                        ;*ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP
                                        ;*BEING TESTED.  OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS
                                        ;*TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY
                                        ;*ON THE SYSTEM.  NOTE THAT HERE THE WORD 'TAG' REFERS
                                        ;*TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21,
                                        ;*WHICH ARE STORED IN THE CACHE ADDRESS MEMORY.  HERE
                                        ;*THE ADDRESS MEMORY IS WRITTEN IN THE UPWARD DIRECTION,
                                        ;*THAT IS 'TAG' 1 IS WRITTEN FIRST, 'TAG' 2 SECOND ETC.
                                        ;*THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED
                                        ;* TO SEE IF IT IS A HIT, THUS MAKING SURE NO
                                        ;*'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER
                                        ;*'TAG'.  NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL
                                        ;*ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD
                                        ;*INVOLVE WRITTING THE 'TAGS' IN THE DOWNWARD DIRECTION
                                        ;*AS WELL AS THE UPWARD DIRECTION.  THE DOWNWARD
                                        ;*WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND
                                        ;*IN TST17.
                                        ;*
                                        ;;*************************************************************
016540  000004                          TST16:  SCOPE
016542  012737  000002  001702                  MOV     #2,$TIMES       ;;DO 2 ITERATIONS
        000016                          UU=$TN-1
016550                                  UU0:
                                                                        ;SET THE SKAD RFGISTER
016550  012737  020162  055572                  MOV     #TST17,SKAD     ;IN CASE THE TEST ABORTS.

016556  012737  055440  000114                  MOV     #SPUR,@#CACHVEC ;AT FIRST EXPECT NO ERRORS
016564  113737  001502  001632                  MOVB    $TSTNM,$TMPO
016572  005037  017650                          CLR     UUFLG3          ;ERROR FLAG.
016576  104416                                  MMSKIP

016600  104417                                  SIZE
016602  000000                          UULOAD: .WORD   0               ;ADDRESS OF THE HIGHEST WORD
016604  000000                          UUHIAD: .WORD   0               ;IN MEMORY

016606  005037  017644                          CLR     UUFLG1          ;TEST GROUP 0 FIRST.
016612  012737  000034  017666                  MOV     #S0M0M1,UUGS
016620  012737  000054  017670                  MOV     #S1M0M1,UUGM

016626  005037  017646                  UU1:    CLR     UUFLG2          ;CLEAR THE PROGRESS FLAG.
016632  012700  016550                          MOV     #UU0,R0         ;MAKE THIS CODE HITS, IN
016636  012701  001000                          MOV     #1000,R1        ;THE GROUP NOT BEING TESTED.

016642  013737  017666  177746  UU2:    MOV     UUGS,@#CONTRL
016650  005760  002000                          TST     2000(R0)
016654  013737  017670  177746                  MOV     UUGM,@#CONTRL
016662  005720                                  TST     (R0)+
016664  077112                                  SOB     R1,UU2
```

```

016666  013701  017666                      MOV     UUGS,R1             ;SELECT THE GROUP BEING TESTED.
016672  042701  177717                      BIC     #177717,R1
016676  010137  177746                      MOV     R1,@#CONTRL


016702  012700  172340                      MOV     #KIPAR0,R0          ;INITIALLY PUT MEMORY
016706  012701  077406                      MOV     #77406,R1           ;MANAGEMENT IN A 'PASSIVE'
016712  012702  172300                      MOV     #KIPDR0,R2          ;STATE, THAT IS MAP ALL
016716  012703  000010                      MOV     #10,R3              ;VIRTUAL ADDRESSES ON TO
016722  010122                      64$:    MOV     R1,(R2)+            ;THEMSELVES AS PHYSICAL
016724  077302                              SOB     R3,64$              ;ADDRESSES.
016726  005020                              CLR     (R0)+
016730  012720  000200                      MOV     #200,(R0)+
016734  012720  000400                      MOV     #400,(R0)+
016740  012720  000600                      MOV     #600,(R0)+
016744  012720  001000                      MOV     #1000,(R0)+
016750  012720  001200                      MOV     #1200,(R0)+
016754  012720  001400                      MOV     #1400,(R0)+
016760  012710  177600                      MOV     #177600,(R0)

016764  012737  000020  172516              MOV     #20,@#MMR3          ;TURN ON MEMORY MANAGEMENT.
016772  012737  000001  177572              MOV     #1,@#MMR0

017000  005037  017656                      CLR     UUADR2              ;INITIALIZE THE ADDRESSES.
017004  005037  017660                      CLR     UUADR2+2
017010  012737  140000  017652              MOV     #140000,UUADR1
017016  005037  017654                      CLR     UUADR1+2
017022  012701  000400                      MOV     #400,R1             ;COUNTER.
017026  012737  017674  000114              MOV     #UUERR1,@#CACHVEC
017034  012737  000001  017646              MOV     #1,UUFLG2           ;KEEP TRACK OF TEST PROGRESS.
017042                              UU3:
                                    ;DOUBLE PRECISION ADDITION, UNSIGNED
017042  013737  017652  017662              MOV     UUADR1,UUADR3
017050  013737  017654  017664              MOV     UUADR1+2,UUADR3+2
017056  063737  017656  017662              ADD     UUADR2,UUADR3
017064  005537  017664                      ADC     UUADR3+2
017070  063737  017660  017664              ADD     UUADR2+2,UUADR3+2




017076                              UU4:

                                    ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
017076  023737  017664  016604              CMP     UUADR3+2,UULOAD+2       ;COMPARE THE HIGH ORDER
017104  001006                              BNE     64$                 ;PARTS OF UUADR3 AND ARG2.
017106  023737  017662  016602              CMP     UUADR3,UULOAD       ;COMPARE THE LOW ORDER

017114  001002                              BNE     64$                 ;PARTS.



017116  000137  017150                      JMP     UU6                 ;THEY WERE EQUAL.

017122  103402                      64$:    BLO     65$
```

```
017124  000137  017134                   JMP     UU5               ;THE FIRST ADDRESS IS LARGER
                                                                   ;THAN THE SECOND!
017130  000137  017150           65$:    JMP     UU6               ;THE FIRST IS LESS THAN THE
                                                                   ;SECOND.


017134  012737  140000  017652   UU5:    MOV     #140000,UUADR1    ;RESET TO GET VALID ADDRESS.
01/142  005037  017654                   CLR     UUADR1+2
017146  000735                           BR      UU3

017150  012702  017662           UU6:    MOV     #UUADR3,R2

017154  011203                           MOV     (R2),R3           ;GET THE PHYSICAL ADDRESS POINTED
017156  042703  177700                   BIC     #177700,R3        ;TO BY R2 AND ESTABLISH
017162  011205                           MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
017164  016204  000002                   MOV     2(R2),R4          ;WILL RELOCATE THROUGH
017170  073427  177772                   ASHC    #-6,R4            ;KIPAR6. SETUP KIPAR6 AND
017174  010537  172354                   MOV     R5,@#KIPAR6       ;LEAVE THE VIRTUAL ADDRESS
017200  062703  140000                   ADD     #140000,R3        ;IN R3.


017204  005713                           TST     (R3)              ;GET A HIT AT THE TEST
017206  005713                           TST     (R3)              ;ADDRESS.

017210  032737  000010  177752           BIT     #10,@#HITMIS
017216  001012                           BNE     UU7
017220  013737  017644  001636           MOV     UUFLG1,$TMP2
017226  013737  017662  001640           MOV     UUADR3,$TMP3
017234  013737  017664  001642           MOV     UUADR3+2,$TMP4
017242  104041                   1$:     ERROR   41
017244  062737  002000  017652   UU7:    ADD     #2000,UUADR1
017252  005537  017654                   ADC     UUADR1+2
017256  062737  000004  017656           ADD     #4,UUADR2         ;LOOP TO WRITE NEXT ADDRESS
017264  005301                           DEC     R1
017266  001402                           BEQ     1$
017270  000137  017042                   JMP     UU3
017274  012737  000002  017646   1$:     MOV     #2,UUFLG2

017302  013700  017670                   MOV     UUGM,R0           ;FROM NOW ON SELECT THE
017306  042700  177717                   BIC     #177717,R0        ;GROUP NOT BEING TESTED.
017312  010037  177746                   MOV     R0,@#CONTRL

017316  005037  017656           UU8:    CLR     UUADR2            ;NOW RE-GENERATE ALL THE
017322  005037  017660                   CLR     UUADR2+2          ;ADDRESS WHICH WERE MADE
017326  012737  140000  017652           MOV     #140000,UUADR1    ;HITS, ABOVE, AND MAKE SURE
017334  005037  017654                   CLR     UUADR1+2          ;THEY ARE STILL HITS.
017340  012701  000400                   MOV     #400,R1
017344  012737  000003  017646           MOV     #3,UUFLG2
017352                           UU9:
                                 ;DOUBLE PRECISION ADDITION, UNSIGNED
017352  013737  017652  017662           MOV     UUADR1,UUADR3
017360  013737  017654  017664           MOV     UUADR1+2,UUADR3+2
017366  063737  017656  017662           ADD     UUADR2,UUADR3
017374  005537  017664                   ADC     UUADR3+2
017400  063737  017660  017664           ADD     UUADR2+2,UUADR3+2

```

```



017406                          UU10:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
017406  023737  017664  016604          CMP     UUADR3+2,UULOAD+2      ;COMPARE THE HIGH ORDER
017414  001006                          BNE     64$               ;PARTS OF UUADR3 AND ARG2.
017416  023737  017662  016602          CMP     UUADR3,UULOAD     ;COMPARE THE LOW ORDER

017424  001002                          BNE     64$               ;PARTS.



017426  000137  017460                  JMP     UU12              ;THEY WERE EQUAL.

017432  103402                  64$:    BLO     65$
017434  000137  017444                  JMP     UU11              ;THE FIRST ADDRESS IS LARGER
                                                                  ;THAN THE SECOND!
017440  000137  017460          65$:    JMP     UU12              ;THE FIRST IS LESS THAN THE
                                                                  ;SECOND.


017444  012737  140000  017652  UU11:   MOV     #140000,UUADR1    ;RESET TO GET A VALID ADDRESS.
017452  005037  017654                  CLR     UUADR1+2
017456  000735                          BR      UU9

017460  012702  017662          UU12:   MOV     #UUADR3,R2

017464  011203                          MOV     (R2),R3           ;GET THE PHYSICAL ADDRESS POINTED
017466  042703  177700                  BIC     #177700,R3        ;TO BY R2 AND ESTABLISH
017472  011205                          MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
017474  016204  000002                  MOV     2(R2),R4          ;WILL RELOCATE THROUGH
017500  073427  177772                  ASHC    #-6,R4            ;KIPAR6. SETUP KIPAR6 AND
017504  010537  172354                  MOV     R5,@#KIPAR6       ;LEAVE THE VIRTUAL ADDRESS
017510  062703  140000                  ADD     #140000,R3        ;IN R3.


017514  005713                          TST     (R3)    ;STILL A HIT?
017516  032737  000010  177752          BIT     #10,@#HITMIS
017524  001012                          BNE     UU13
                                                                  ;NOT A HIT, A DUAL ADDRESSING
017526  013737  017644  001636          MOV     UUFLG1,$TMP2      ;PROBLEM?
017534  013737  017662  001640          MOV     UUADR3,$TMP3
017542  013737  017664  001642          MOV     UUADR3+2,$TMP4
017550  104042                  1$:     ERROR   42

017552  062737  002000  017652  UU13:   ADD     #2000,UUADR1
017560  005537  017654                  ADC     UUADR1+2
017564  062737  000004  017656          ADD     #4,UUADR2         ;LOOP TO READ NEXT ADDRESS
017572  005301                          DEC     R1
017574  001402                          BEQ     1$
017576  000137  017352                  JMP     UU9
017602  012737  000004  017646  1$:     MOV     #4,UUFLG2
017610  005737  017644          UU14:   TST     UUFLG1  ;TESTED BOTH GROUPS?
017614  001161                          BNE     UUDONE            ;YES.
```

```
017616  012737  000001  017644            MOV     #1,UUFLG1          ;NO, GO TEST GROUP 1.
017624  012737  000054  017666            MOV     #S1MOM1,UUGS
017632  012737  000034  017670            MOV     #SOMOM1,UUGM
017640  000137  016626                    JMP     UU1

017644  000000                    UUFLG1: .WORD   0                  ;FLAG WHICH DESIGNATES
                                                                     ;WHICH GROUP IS BEING TESTED,
                                                                     ;1 OR 0.
017646  000000                    UUFLG2: .WORD   0                  ;DESIGNATES HOW FAR THE
                                                                     ;TEST HAS PROGRESSED.
017650  000000                    UUFLG3: .WORD   0                  ;ERROR DURING TEST UUFLG2=4
                                                                     ;PHASE.
017652  000000                    UUADR1: .WORD   0                  ;ADDRESS WRITTEN INTO ADDRESS
017654  000000                            .WORD   0                  ;MEMORY LOCATION
017656  000000                    UUADR2: .WORD   0                  ;ADDRESS MEMORY LOCATION
017660  000000                            .WORD   0                  ;BEING TESTED
017662  000000                    UUADR3: .WORD   0                  ;TEST ADDRESS:UUADR3=UUADR1+UUADR2
017664  000000                            .WORD   0

017666  000000                    UUGS:   .WORD   0                  ;PATTERNS FOR THE CACHE CONTROL
017670  000000                    UUGM:   .WORD   0                  ;REGISTER.
017672  000000                    UUTMP:  .WORD   0

017674  032737  000060  177744    UUERR1: BIT     #60,@#MEMERR       ;WAS THE ERROR A CACHE ADDRESS
017702  001002                            BNE     UUERR2             ;MEMORY PARITY ERROR
017704  000137  055440                    JMP     SPUR

017710                            UUERR2:                            ;REPORT ERROR.
017710  012637  001636                    MOV     (SP)+,$TMP2
017714  005726                            TST     (SP)+
017716  013737  017644  001640            MOV     UUFLG1,$TMP3
017724  013737  177744  001642            MOV     @#MEMERR,$TMP4
017732  013737  017662  001644            MOV     UUADR3,$TMP5
017740  013737  017664  001646            MOV     UUADR3+2,$TMP6
017746  013737  177740  001650            MOV     @#LOADRS,$TMP7
017754  013737  177742  001652            MOV     @#HIADRS,$TMP10
017762  104043                    1$:     ERROR   43

017764  042737  177717  001642            BIC     #177717,$TMP4      ;TRY TO GET THE BAD ADDRESS
017772  013737  177746  017672            MOV     @#CONTRL,UUTMP     ;OUT OF THE ADDRESS MEMORY.
020000  012737  020030  000114            MOV     #UUERR3,@#CACHVEC
020006  013705  177740                    MOV     @#LOADRS,R5
020012  042705  176001                    BIC     #176001,R5
020016  013737  001642  177746            MOV     $TMP4,@#CONTRL
020024  005715                            TST     (R5)
020026  000401                            BR      UUERR4
020030  022626                    UUERR3: CMP     (SP)+,(SP)+
020032  012737  177777  177744    UUERR4: MOV     #-1,@#MEMERR

020040  013737  017672  177746            MOV     UUTMP,@#CONTRL     ;RESET THE CONTROL REGISTER.
020046  012737  017674  000114            MOV     #UUERR1,@#CACHVEC

020054  023727  017646  000001            CMP     UUFLG2,#1          ;RETURN, USING UUFLG2 TO
020062  001002                            BNE     1$                 ;DECIDE WHERE.
020064  000137  017244                    JMP     UU7
020070  023727  017646  000002    1$:     CMP     UUFLG2,#2
```

```
020076  001002                           BNE     2$
020100  000137  017316                   JMP     UU8
020104  023727  017646  000003  2$:      CMP     UUFLG2,#3
020112  001002                           BNE     3$
020114  000137  017552                   JMP     UU13
020120  023727  017646  000004  3$:      CMP     UUFLG2,#4
020126  001007                           BNE     4$
020130  005737  017650                   TST     UUFLG3
020134  001011                           BNE     UUDONE
020136  005337  017650                   DEC     UUFLG3
020142  000137  017610                   JMP     UU14

020146  005737  017646          4$:      TST     UUFLG2
020152  001002                           BNE     UUDONE          ;??HALT???
020154  000137  016626                   JMP     UU1

020160  104414                  UUDONE·RSET                      ;DONE.



                                ;;*****************************************************************
                                ;*TEST 17        CACHE ADDRESS MEMORY DUAL ADDRESS TEST, DOWNWARD
                                ;*
                                ;*THIS IS A DUAL ADDRESS TEST OF THE CACHE ADDRESS
                                ;*MEMORY.  AS MANY AS POSSIBLE DIFFERENT ADDRESS 'TAGS'
                                ;*ARE STORED IN THE 256 (DEC) ADDRESS LOCATIONS OF THE GROUP
                                ;*BEING TESTED.  OBVIOUSLY THE NUMBER OF DIFFERENT ADDRESS
                                ;*TAGS AVAILABLE IS LIMITED BY THE SIZE OF THE MEMORY
                                ;*ON THE SYSTEM.  NOTE THAT HERE THE WORD 'TAG' REFERS
                                ;*TO THAT PART OF AN ADDRESS, BITS 10 THROUGH 21,
                                ;*WHICH ARE STORED IN THE CACHE ADDRESS MEMORY.  HERE
                                ;*THE ADDRESS MEMORY IS WRITTEN IN THE DOWNWARD DIRECTION,
                                ;*THAT IS 'TAG' 256 IS WRITTEN FIRST, 'TAG' 255 SECOND ETC.
                                ;*THEN EACH ADDRESS WHICH WAS WRITTEN IS TESTED
                                ;* TO SEE IF IT IS A HIT, THUS MAKING SURE NO
                                ;*'TAG' WAS OVERWRITTEN BY A REFERENCE TO ANOTHER
                                ;*'TAG'.  NOTE THAT THIS DOES NOT PERFORM A COMPLETE DUAL
                                ;*ADDRESS TEST ON THE ADDRESS MEMORY, FOR THAT WOULD
                                ;*INVOLVE WRITTING THE 'TAGS' IN THE UPWARD DIRECTION
                                ;*AS WELL AS THE DOWNWARD DIRECTION.  THE UPWARD
                                ;*WRITING PART OF THIS DUAL ADDRESS TEST IS FOUND
                                ;*IN TST16.
                                ;*
                                ;;*****************************************************************
020162  000004                  TST17:   SCOPE
020164  012737  000002  001702           MOV     #2,$TIMES       ;;DO 2 ITERATIONS
        000017                  VV=$TN-1
020172                          VV0:
                                                                 ;SET THE SKAD REGISTER
020172  012737  021610  055572           MOV     #TST20,SKAD     ;IN CASE THE TEST ABORTS.

020200  012737  055440  000114           MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.
020206  113737  001502  001632           MOVB    $TSTNM,$TMPO

020214  005037  021276                   CLR     VVFLG3          ;CLEAR THE ERROR FLAG.

020220  104416                           MMSKIP
```

```

020222 104417                              SIZE
020224 000000                     VVLOAD:  .WORD    0                  ;ADDRESS OF THE HIGHEST
020226 000000                     VVHIAD:  .WORD    0                  ;WORD IN MEMORY.

020230 005037 021272                       CLR      VVFLG1             ;TEST GROUP 0 FIRST
020234 012737 000034 021314                MOV      #S0M0M1,VVGS
020242 012737 000054 021316                MOV      #S1M0M1,VVGM

020250 005037 021274              VV1:     CLR      VVFLG2             ;CLEAR THE PROGRESS FLAG
020254 012700 020172                       MOV      #VV0,R0            ;MAKE THIS CODE HITS IN
020260 012701 001000                       MOV      #1000,R1           ;THE GROUP NOT BEING
                                                                       ;TESTED.
020264 013737 021314 177746       VV2:     MOV      VVGS,@#CONTRL
020272 005760 002000                       TST      2000(R0)
020276 013737 021316 177746                MOV      VVGM,@#CONTRL
020304 005720                              TST      (R0)+
020306 077112                              SOB      R1,VV2

020310 013700 021314                       MOV      VVGS,R0            ;FROM NOW ON SELECT
020314 042700 177717                       BIC      #177717,R0         ;THE GROUP BEING TESTED.
020320 010037 177746                       MOV      R0,@#CONTRL


020324 012700 172340                       MOV      #KIPAR0,R0         ;INITIALLY PUT MEMORY
020330 012701 077406                       MOV      #77406,R1          ;MANAGEMENT IN A 'PASSIVE'
020334 012702 172300                       MOV      #KIPDR0,R2         ;STATE, THAT IS MAP ALL
020340 012703 000010                       MOV      #10,R3             ;VIRTUAL ADDRESSES ON TO
020344 010122                     64$:     MOV      R1,(R2)+           ;THEMSELVES AS PHYSICAL
020346 077302                              SOB      R3,64$             ;ADDRESSES.
020350 005020                              CLR      (R0)+
020352 012720 000200                       MOV      #200,(R0)+
020356 012720 000400                       MOV      #400,(R0)+
020362 012720 000600                       MOV      #600,(R0)+
020366 012720 001000                       MOV      #1000,(R0)+
020372 012720 001200                       MOV      #1200,(R0)+
020376 012720 001400                       MOV      #1400,(R0)+
020402 012710 177600                       MOV      #177600,(R0)

020406 012737 000020 172516                MOV      #20,@#MMR3         ;TURN ON MEMORY MANAGEMENT.
020414 012737 000001 177572                MOV      #1,@#MMR0

020422 012737 001774 021304                MOV      #1774,VVADR2       ;INITIALIZE THE ADDRESSES
020430 005037 021306                       CLR      VVADR2+2
020434 012737 140000 021300                MOV      #140000,VVADR1
020442 005037 021302                       CLR      VVADR1+2
020446 012701 000400                       MOV      #400,R1            ;A COUNTER.
020452 012737 021322 000114                MOV      #VVERR1,@#CACHVEC          ;EXPECT ERRORS NOW.
020460 012737 000001 021274                MOV      #1,VVFLG2          ;KEEP TRACK OF TEST PROGRESS.

020466                            VV3:
                                  ;DOUBLE PRECISION ADDITION, UNSIGNED
020466 013737 021300 021310                MOV      VVADR1,VVADR3
020474 013737 021302 021312                MOV      VVADR1+2,VVADR3+2
020502 063737 021304 021310                ADD      VVADR2,VVADR3
020510 005537 021312                       ADC      VVADR3+2
```

```
020514  063737  021306  021312          ADD     VVADR2+2,VVADR3+2




020522                          VV4:

                                ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
020522  023737  021312  020226          CMP     VVADR3+2,VVLOAD+2    ;COMPARE THE HIGH ORDER
020530  001006                          BNE     64$             ;PARTS OF VVADR3 AND ARG2.
020532  023737  021310  020224          CMP     VVADR3,VVLOAD   ;COMPARE THE LOW ORDER

020540  001002                          BNE     64$             ;PARTS.



020542  000137  020574                  JMP     VV6             ;THEY WERE EQUAL!

020546  103402                  64$:    BLO     65$
020550  000137  020560                  JMP     VV5             ;THE FIRST ADDRESS IS LARGER
                                                                ;THAN THE SECOND!
020554  000137  020574          65$:    JMP     VV6             ;THE FIRST IS LESS THAN THE
                                                                ;SECOND.



020560  012737  140000  021300  VV5:    MOV     #140000,VVADR1  ;RESET TO GET A VALID ADDRESS.
020566  005037  021302                  CLR     VVADR1+2
020572  000735                          BR      VV3

020574  012702  021310          VV6:    MOV     #VVADR3,R2

020600  011203                          MOV     (R2),R3         ;GET THE PHYSICAL ADDRESS POINTED
020602  042703  177700                  BIC     #177700,R3      ;TO BY R2 AND ESTABLISH
020606  011205                          MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
020610  016204  000002                  MOV     2(R2),R4        ;WILL RELOCATE THROUGH
020614  073427  177772                  ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
020620  010537  172354                  MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
020624  062703  140000                  ADD     #140000,R3      ;IN R3.


020630  005713                          TST     (R3)            ;GET A HIT AT THE
020632  005713                          TST     (R3)            ;TEST ADDRESS
020634  032737  000010  177752          BIT     #10,@#HITMIS
020642  001012                          BNE     VV7
                                                                ;REPORT FAILURE TO GET A HIT.
020644  013737  021272  001636          MOV     VVFLG1,$TMP2
020652  013737  021310  001640          MOV     VVADR3,$TMP3
020660  013737  021312  001642          MOV     VVADR3+2,$TMP4
020666  104041                  1$:     ERROR   41

020670  062737  002000  021300  VV7:    ADD     #2000,VVADR1
020676  005537  021302                  ADC     VVADR1+2
020702  062737  177774  021304          ADD     #-4,VVADR2      ;LOOP TO WRITE NEXT ADDRESS
020710  005301                          DEC     R1
020712  001402                          BEQ     1$
```

```
020714  000137  020466                          JMP     VV3
020720  012737  000002  021274  1$:             MOV     #2,VVFLG2

020726  013700  021316                          MOV     VVGM,R0         ;FROM NOW ON SELECT
020732  042700  177717                          BIC     #177717,R0      ;THE GROUP NOT BEING
020736  010037  177746                          MOV     R0,@#CONTRL     ;TESTED.

020742  012737  001774  021304  VV8:            MOV     #1774,VVADR2    ;NOW RE-GENERATE ALL THE
020750  005037  021306                          CLR     VVADR2+2        ;ADDRESSES MADE HITS IN
020754  012737  140000  021300                  MOV     #140000,VVADR1  ;THE ABOVE PORTION OF
020762  005037  021302                          CLR     VVADR1+2        ;THE TEST, AND MAKE SURE
020766  012701  000400                          MOV     #400,R1         ;THEY ARE STILL HITS.
020772  012737  000003  021274                  MOV     #3,VVFLG2
021000                                  VV9:
                                        ;DOUBLE PRECISION ADDITION, UNSIGNED
021000  013737  021300  021310                  MOV     VVADR1,VVADR3
021006  013737  021302  021312                  MOV     VVADR1+2,VVADR3+2
021014  063737  021304  021310                  ADD     VVADR2,VVADR3
021022  005537  021312                          ADC     VVADR3+2
021026  063737  021306  021312                  ADD     VVADR2+2,VVADR3+2




021034                                  VV10:

                                        ;DOUBLE PRECISION COMPARE OF TWO 22-BIT ADDRESSES
021034  023737  021312  020226                  CMP     VVADR3+2,VVLOAD+2       ;COMPARE THE HIGH ORDER
021042  001006                                  BNE     64$             ;PARTS OF VVADR3 AND ARG2.
021044  023737  021310  020224                  CMP     VVADR3,VVLOAD   ;COMPARE THE LOW ORDER

021052  001002                                  BNE     64$             ;PARTS.



021054  000137  021106                          JMP     VV12            ;THEY WERE EQUAL!

021060  103402                          64$:    BLO     65$
021062  000137  021072                          JMP     VV11            ;THE FIRST ADDRESS IS LARGER
                                                                        ;THAN THE SECOND!
021066  000137  021106                  65$:    JMP     VV12            ;THE FIRST IS LESS THAN THE
                                                                        ;SECOND.


021072  012737  140000  021300  VV11:           MOV     #140000,VVADR1  ;RESET TO CREATE A VALID
021100  005037  021302                          CLR     VVADR1+2        ;ADDRESS
021104  000735                                  BR      VV9

021106  012702  021310                  VV12:   MOV     #VVADR3,R2

021112  011203                                  MOV     (R2),R3         ;GET THE PHYSICAL ADDRESS POINTED
021114  042703  177700                          BIC     #177700,R3      ;TO BY R2 AND ESTABLISH
021120  011205                                  MOV     (R2),R5 ;A VIRTUAL ADDRESS WHICH
021122  016204  000002                          MOV     2(R2),R4        ;WILL RELOCATE THROUGH
021126  073427  177772                          ASHC    #-6,R4          ;KIPAR6. SETUP KIPAR6 AND
021132  010537  172354                          MOV     R5,@#KIPAR6     ;LEAVE THE VIRTUAL ADDRESS
```

```
021136  062703  140000                    ADD     #140000,R3       ;IN R3.


021142  005713                            TST     (R3)             ;STILL A HIT?
021144  032737  000010  177752            BIT     #10,@#HITMIS
021152  001012                            BNE     VV13
                                                                   ;REPORT ERROR.
021154  013737  021272  001636            MOV     VVFLG1,$TMP2
021162  013737  021310  001640            MOV     VVADR3,$TMP3
021170  013737  021312  001642            MOV     VVADR3+2,$TMP4
021176  104042                    1$:     ERROR   42

021200  062737  002000  021300  VV13:     ADD     #2000,VVADR1
021206  005537  021302                    ADC     VVADR1+2
021212  062737  177774  021304            ADD     #-4,VVADR2
021220  005301                            DEC     R1
021222  001402                            BEQ     1$
021224  000137  021000                    JMP     VV9
021230  012737  000004  021274  1$:       MOV     #4,VVFLG2
021236  005737  021272          VV14:     TST     VVFLG1           ;TESTED BOTH GROUPS?
021242  001161                            BNE     VVDONE           ;YES.
021244  012737  000034  021316            MOV     #S0M0M1,VVGM     ;NO GO TEST GROUP 1.
021252  012737  000054  021314            MOV     #S1M0M1,VVGS
021260  012737  000001  021272            MOV     #1,VVFLG1
021266  000137  020250                    JMP     VV1

021272  000000                  VVFLG1:  .WORD    0                ;0 OR 1, GROUP BEING TESTED.
021274  000000                  VVFLG2:  .WORD    0                ;TEST PROGRESS FLAG.
021276  000000                  VVFLG3:  .WORD    0                ;ERROR FLAG.

021300  000000                  VVADR1:  .WORD    0                ;PATTERN WRITTEN INTO THE ADDRESS
021302  000000                           .WORD    0                ;MEMORY LOCATION.
021304  000000                  VVADR2:  .WORD    0                ;ADDRESS MEMORY LOCATION BEING
021306  000000                           .WORD    0                ;TESTED X 4.
021310  000000                  VVADR3:  .WORD    0                ;TEST ADDRESS.
021312  000000                           .WORD    0                ;VVADR3=VVADR2+VVADR1

021314  000000                  VVGS:    .WORD    0                ;PATTERNS FOR THE CACHE
021316  000000                  VVGM:    .WORD    0                ;CONTROL REGISTER.

021320  000000                  VVTMP:   .WORD    0

021322  032737  000060  177744  VVERR1:  BIT      #60,@#MEMERR     ;WAS THE ERROR THAT CAUSED
021330  001002                           BNE      VVERR2           ;THE TRAP TO HERF A CACHE
021332  000137  055440                   JMP      SPUR             ;ADDRESS MEMORY PARITY ERROR?

021336                          VVERR2:                            ;REPORT ERROR.
021336  012637  001636                   MOV      (SP)+,$TMP2
021342  005726                           TST      (SP)+
021344  013737  021272  001640           MOV      VVFLG1,$TMP3
021352  013737  177744  001642           MOV      @#MEMERR,$TMP4
021360  013737  021310  001644           MOV      VVADR3,$TMP5
021366  013737  021312  001646           MOV      VVADR3+2,$TMP6
021374  013737  177740  001650           MOV      @#LOADRS,$TMP7
021402  013737  177742  001652           MOV      @#HIADRS,$TMP10
021410  104043                  1$:      ERROR    43
```

```

021412  042737  177717  001642          BIC     #177717,$TMP4    ;TRY TO GET THE BAD ADDRESS
021420  013737  177746  021320          MOV     @#CONTRL,VVTMP   ;OUT OF THE ADDRESS MEMORY.
021426  012737  021456  000114          MOV     #VVERR3,@#CACHVEC
021434  013705  177740                  MOV     @#LOADRS,R5
021440  042705  176001                  BIC     #176001,R5
021444  013737  001642  177746          MOV     $TMP4,@#CONTRL
021452  005715                          TST     (R5)
021454  000401                          BR      VVERR4
021456  022626                  VVERR3: CMP     (SP)+,(SP)+
021460  012737  177777  177744  VVEPR4: MOV     #-1,@#MEMERR

021466  013737  021320  177746          MOV     VVTMP,@#CONTRL   ;RESET THE CONTRL REGISTER
021474  012737  021322  000114          MOV     #VVERR1,@#CACHVEC
021502  023727  021274  000001          CMP     VVFLG2,#1        ;RETURN, USING VVFLG2 TO
021510  001002                          BNE     1$               ;DECIDE WHERE.
021512  000137  020670                  JMP     VV7
021516  023727  021274  000002  1$:     CMP     VVFLG2,#2
021524  001002                          BNE     2$
021526  000137  020742                  JMP     VV8
021532  023727  021274  000003  2$:     CMP     VVFLG2,#3
021540  001002                          BNE     3$
021542  000137  021200                  JMP     VV13
021546  023727  021274  000004  3$:     CMP     VVFLG2,#4
021554  001007                          BNE     4$
021556  005737  021276                  TST     VVFLG3
021562  001011                          BNE     VVDONE
021564  005337  021276                  DEC     VVFLG3
021570  000137  021236                  JMP     VV14
021574  005737  021274          4$:     TST     VVFLG2
021600  001002                          BNE     VVDONE           ;????HALT???
021602  000137  020250                  JMP     VV1

021606  104414                  VVDONE: RSET                             ;DONE!


                                ;;*****************************************************************
                                ;*TEST 20        CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ONES TEST
                                ;*
                                ;*THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC.  THIS
                                ;*IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING
                                ;*A WRITE.  IT DESIGNATES WHICH BYTES OF THE TWO WORDS OF
                                ;*DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO
                                ;*BE WRITTEN.  THIS WILL BE A TEST DOING CPU DATOB REFERENCES TO
                                ;*THE CACHE.  THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN
                                ;*OF ZEROES.
                                ;*
                                ;;*****************************************************************
021610  000004                  TST20:  SCOPE
021612  012737  000010  001702          MOV     #10,$TIMES       ;;DO 10 ITERATIONS
        000020                  CC=$TN-1
                                                                 ;SET THE SKAD REGISTER
021620  012737  022374  055572          MOV     #TST21,SKAD      ;IN CASE THE TEST ABORTS.

021626  113737  001502  001632          MOVB    $TSTNM,$TMP0
021634  012737  022100  000114          MOV     #CCERR1,@#CACHVEC
```

```

021642  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES

021650  012700  022074                  MOV     #CCTMP2,R0      ;LOCATE THE TEST SPACE.
021654  042700  000003                  BIC     #3,R0
021660  010001                          MOV     R0,R1
021662  005010                  CC1:    CLR     (R0)            ;TEST MASK 0
021664  005060  000002                  CLR     2(R0)
021670  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
021672  112711  000377                  MOVB    #377,(R1)
021676  022710  000377                  CMP     #377,(R0)
021702  001403                          BEQ     CC3
021704  004737  022312          CC2:    JSR     PC,CCERR3
021710  000403                          BR      CC4
021712  005760  000002          CC3:    TST     2(R0)
021716  001372                          BNE     CC2
021720  062701  000001          CC4:    ADD     #1,R1           ;TEST MASK 1.
021724  005010                          CLR     (R0)
021726  005060  000002                  CLR     2(R0)
021732  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
021734  112711  000377                  MOVB    #377,(R1)
021740  022710  177400                  CMP     #177400,(R0)
021744  001403                          BEQ     CC6
021746  004737  022312          CC5:    JSR     PC,CCERR3
021752  000403                          BR      CC7
021754  005760  000002          CC6:    TST     2(R0)
021760  001372                          BNE     CC5

021762  062701  000001          CC7:    ADD     #1,R1           ;TEST MASK 2.
021766  005010                          CLR     (R0)
021770  005060  000002                  CLR     2(R0)
021774  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
021776  112711  000377                  MOVB    #377,(R1)
022002  022760  000377  000002          CMP     #377,2(R0)
022010  001403                          BEQ     CC9
022012  004737  022312          CC8:    JSR     PC,CCERR3
022016  000402                          BR      CC10
022020  005710                  CC9:    TST     (R0)
022022  001373                          BNE     CC8

022024  062701  000001          CC10:   ADD     #1,R1           ;TEST MASK 3.
022030  005010                          CLR     (R0)
022032  005060  000002                  CLR     2(R0)
022036  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
022040  112711  000377                  MOVB    #377,(R1)
022044  022760  177400  000002          CMP     #177400,2(R0)
022052  001403                          BEQ     CC12
022054  004737  022312          CC11:   JSR     PC,CCERR3
022060  000402                          BR      CC13
022062  005710                  CC12:   TST     (R0)
022064  001373                          BNE     CC11

022066  000137  022372          CC13:   JMP     CCDONE

022072  000000                  CCTMP1: .WORD   0
022074  000000                  CCTMP2: .WORD   0               ;THE TEST AREA.
```

```
022076  000000                                  .WORD   0


022100  032737  000002  177744  CCERR1: BIT     #2,@#MEMERR     ;SHOULD BE A MAIN MEMORY
022106  001002                          BNE     1$              ;ADDRESS AND CONTROL LINE
022110  000137  055440                  JMP     SPUR            ;PARITY ERROR.
022114  020137  177740          1$:     CMP     R1,@#LOADRS     ;ERROR ADDRESS SHOULD BE
022120  001402                          BEQ     CCERR2          ;TEST ADDRESS.
022122  000137  055440                  JMP     SPUR
022126  012637  001646          CCERR2: MOV     (SP)+,$TMP6
022132  005037  001670                  CLR     $TMP17
022136  005726                          TST     (SP)+           ;RESET THE STACK
022140  012737  000044  001672          MOV     #44,$TMP20
022146  013737  177740  001640          MOV     @#LOADRS,$TMP3
022154  013737  177742  001642          MOV     @#HIADRS,$TMP4
022162  013737  177744  001644          MOV     @#MEMERR,$TMP5
022170  010037  001646                  MOV     R0,$TMP6
022174  005037  001650                  CLR     $TMP7
022200  010037  001662                  MOV     R0,$TMP14
022204  062737  000002  001662          ADD     #2,$TMP14
022212  005037  001664                  CLR     $TMP15
022216  011037  001652                  MOV     (R0),$TMP10
022222  016037  000002  001654          MOV     2(R0),$TMP11
022230  010137  001656                  MOV     R1,$TMP12
022234  005037  001660                  CLR     $TMP13
022240  104044                  64$:    ERROR   44
022242  012737  177777  177744          MOV     #-1,@#MEMERR

022250  010002                          MOV     R0,R2
022252  020102                          CMP     R1,R2
022254  001002                          BNE     2$
022256  000137  021720                  JMP     CC4
022262  005202                  2$:     INC     R2
022264  020102                          CMP     R1,R2
022266  001002                          BNE     3$
022270  000137  021762                  JMP     CC7
022274  005202                  3$:     INC     R2
022276  020102                          CMP     R1,R2
022300  001002                          BNE     4$
022302  000137  022024                  JMP     CC10
022306  000137  022372          4$:     JMP     CCDONE


022312  011637  001656          CCERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
                                                                ;THE CORRECT BYTE
022316  010037  001636                  MOV     R0,$TMP2
022322  005037  001640                  CLR     $TMP3
022326  010037  001642                  MOV     R0,$TMP4
022332  062737  000002  001642          ADD     #2,$TMP4
022340  005037  001644                  CLR     $TMP5
022344  011037  001646                  MOV     (R0),$TMP6
022350  016037  000002  001650          MOV     2(R0),$TMP7
022356  010137  001652                  MOV     R1,$TMP10
022362  005037  001654                  CLR     $TMP11
022366  104046                          ERROR   46
022370  000207                          RTS     PC
```

```


022372  104414                  CCDONE: RSET                      ;DONE


                                ;;*******************************************************************
                                ;*TEST 21         CACHE ADDRESS MEMORY BYTE MASK GENERATOR, CPU DATOB ZEROES TEST
                                ;*
                                ;*THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC.
                                ;*HERE CPU DATOB'S WILL MOVE ZEROES INTO A BACKROUND
                                ;*PATTERN OF ONES.
                                ;*
                                ;;*******************************************************************
022374  000004                  TST21:  SCOPE
022376  012737  000010  001702          MOV     #10,$TIMES        ;;DO 10 ITERATIONS
        000021                  FF=$TN-1
                                                                  ;SET THE SKAD REGISTER
022404  012737  023172  055572          MOV     #TST22,SKAD       ;IN CASE THE TEST ABORTS.

022412  113737  001502  001632          MOVB    $TSTNM,$TMP0
022420  012737  022676  000114          MOV     #FFERR1,@#CACHVEC

022426  012737  000014  177746          MOV     #MOM1,@#CONTRL    ;FORCE MISSES.

022434  012700  022672                  MOV     #FFTMP2,R0
022440  042700  000003                  BIC     #3,R0
022444  010001                          MOV     R0,R1

022446  012710  177777          FF1:    MOV     #-1,(R0)          ;TEST MASK 0
022452  012760  177777  000002          MOV     #-1,2(R0)
022460  000240                          NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
022462  105011                          CLRB    (R1)
022464  022710  177400                  CMP     #177400,(R0)
022470  001403                          BEQ     FF3
022472  004737  023110          FF2:    JSR     PC,FFERR3
022476  000404                          BR      FF4
022500  022760  177777  000002  FF3:    CMP     #-1,2(R0)
022506  001371                          BNE     FF2

022510  005201                  FF4:    INC     R1                ;TEST MASK 1.
022512  012710  177777                  MOV     #-1,(R0)
022516  012760  177777  000002          MOV     #-1,2(R0)
022524  000240                          NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
022526  105011                          CLRB    (R1)
022530  022710  000377                  CMP     #377,(R0)
022534  001403                          BEQ     FF6
022536  004737  023110          FF5:    JSR     PC,FFERR3
022542  000404                          BR      FF7
022544  022760  177777  000002  FF6:    CMP     #-1,2(R0)
022552  001371                          BNE     FF5

022554  005201                  FF7:    INC     R1                ;TEST MASK 2.
022556  012710  177777                  MOV     #-1,(R0)
022562  012760  177777  000002          MOV     #-1,2(R0)
022570  000240                          NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
022572  105011                          CLRB    (R1)
```

```
022574  022760  177400  000002             CMP     #177400,2(R0)
022602  001403                             BEQ     FF9
022604  004737  023110            FF8:     JSR     PC,FFERR3
022610  000403                             BR      FF10
022612  022710  177777            FF9:     CMP     #-1,(R0)
022616  001372                             BNE     FF8

022620  005201                    FF10:    INC     R1              ;TEST MASK 3.
022622  012710  177777                     MOV     #-1,(R0)
022626  012760  177777  000002             MOV     #-1,2(R0)
022634  000240                             NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
022636  105011                             CLRB    (R1)
022640  022760  000377  000002             CMP     #377,2(R0)
022646  001403                             BEQ     FF12
022650  004737  023110            FF11:    JSR     PC,FFERR3
022654  000403                             BR      FF13
022656  022710  177777            FF12:    CMP     #-1,(R0)
022662  001372                             BNE     FF11

022664  000137  023170            FF13:    JMP     FFDONE

022670  000000                    FFTMP1:  .WORD   0               ;TEST AREA.
022672  000000                    FFTMP2:  .WORD   0
022674  000000                             .WORD   0


022676  032737  000002  177744    FFERR1:  BIT     #2,@#MEMERR     ;SHOULD BE A MAIN MEMORY
022704  001002                             BNE     1$              ;ADDRESS AND CONTROL LINE
022706  000137  055440                     JMP     SPUR            ;PARITY ERROR.
022712  020137  177740            1$:      CMP     R1,@#LOADRS     ;ERROR ADDRESS SHOULD BE
022716  001402                             BEQ     FFERR2          ;TEST ADDRESS.
022720  000137  055440                     JMP     SPUR
022724  012637  001646            FFERR2:  MOV     (SP)+,$TMP6
022730  005037  001670                     CLR     $TMP17
022734  005726                             TST     (SP)+           ;RESET THE STACK
022736  012737  000050  001672             MOV     #50,$TMP20
022744  013737  177740  001640             MOV     @#LOADRS,$TMP3
022752  013737  177742  001642             MOV     @#HIADRS,$TMP4
022760  013737  177744  001644             MOV     @#MEMERR,$TMP5
022766  010037  001646                     MOV     R0,$TMP6
022772  005037  001650                     CLR     $TMP7
022776  010037  001662                     MOV     R0,$TMP14
023002  062737  000002  001662             ADD     #2,$TMP14
023010  005037  001664                     CLR     $TMP15
023014  011037  001652                     MOV     (R0),$TMP10
023020  016037  000002  001654             MOV     2(R0),$TMP11
023026  010137  001656                     MOV     R1,$TMP12
023032  005037  001660                     CLR     $TMP13
023036  104050                    64$:     ERROR   50
023040  012737  177777  177744             MOV     #-1,@#MEMERR

023046  010002                             MOV     R0,R2
023050  020102                             CMP     R1,R2
023052  001002                             BNE     2$
023054  000137  022510                     JMP     FF4
023060  005202                    2$:      INC     R2
```

```
023062  020102                          CMP     R1,R2
023064  001002                          BNE     3$
023066  000137  022554                  JMP     FF7
023072  005202                  3$:     INC     R2
023074  020102                          CMP     R1,R2
023076  001002                          BNE     4$
023100  000137  022620                  JMP     FF10
023104  000137  023170          4$:     JMP     FFDONE          ;HALT????


023110  011637  001656          FFERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
                                                                ;THE CORRECT BYTE
023114  010037  001636                  MOV     R0,$TMP2
023120  005037  001640                  CLR     $TMP3
023124  010037  001642                  MOV     R0,$TMP4
023130  062737  000002  001642          ADD     #2,$TMP4
023136  005037  001644                  CLR     $TMP5
023142  011037  001646                  MOV     (R0),$TMP6
023146  016037  000002  001650          MOV     2(R0),$TMP7
023154  010137  001652                  MOV     R1,$TMP10
023160  005037  001654                  CLR     $TMP11
023164  104052                          ERROR   52
023166  000207                          RTS     PC


023170  104414                  FFDONE: RSET                    ;DONE!

                                ;;*****************************************************************
                                ;*TEST 22       CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ONES TEST
                                ;*
                                ;*THIS IS A TEST OF THE BYTE MASK GENERATION LOGIC.  THIS
                                ;*IS A FOUR BIT MASK USED BY MAIN MEMORY WHEN PERFORMING
                                ;*A WRITE.  IT DESIGNATES WHICH BYTES OF THE TWO WORDS OF
                                ;*DATA ON THE MAIN MEMORY DATA BUS LINES ARE TO
                                ;*BE WRITTEN.  THIS WILL BE A TEST DOING UNIBUS DATOB REFERENCES TO
                                ;*THE CACHE.  THE DATOB WILL WRITE 377 INTO A BACK ROUND PATTERN
                                ;*OF ZEROES.
                                ;*
                                ;;*****************************************************************
023172  000004                  TST22:  SCOPE
023174  012737  000010  001702          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
        000022                  EE=$TN-1
                                                                ;SET THE SKAD REGISTER
023202  012737  024060  055572          MOV     #TST23,SKAD     ;IN CASE THE TEST ABORTS.

023210  113737  001502  001632          MOVB    $TSTNM,$TMP0
023216  104416                          MMSKIP
023220  012737  023564  000114          MOV     #EEERR1,@#CACHVEC

023226  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                ;TO RELOCATE EVERYTHING
023232  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
023236  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
023242  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
023244  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                                ;RELOCATED TO THEMSELVES.
```

```
023250 012722 077406          64$:  MOV   #77406,(R2)+
023254 010401                       MOV   R4,R1
023256 072127 000006                ASH   #6,R1
023262 010125                       MOV   R1,(R5)+
023264 005025                       CLR   (R5)+
023266 010410                       MOV   R4,(R0)
023270 062720 170000                ADD   #170000,(R0)+
023274 062704 000200                ADD   #200,R4
023300 077315                       SOB   R3,64$
023302 012710 177600                MOV   #177600,(R0)
023306 012712 077406                MOV   #77406,(R2)

023312 012737 000060 172516         MOV   #60,@#MMR3       ;TURN ON MEMORY MANAGEMENT
023320 012737 000001 177572         MOV   #1,@#MMR0        ;AND THE MAPPING BOX RELOCATION.

023326 012737 000014 177746         MOV   #MOM1,@#CONTRL   ;FORCE MISSES TO BOTH GROUPS.

023334 012700 023560                MOV   #EETMP2,R0       ;LOCATE THE TEST SPACE.
023340 042700 000003                BIC   #3,R0
023344 010001                       MOV   R0,R1

023346 005010                 EE1:  CLR   (R0)             ;TEST MASK 0
023350 005060 000002                CLR   2(R0)
023354 000240                       NOP                    ;FOR SCOPING WITH AN OSCILLOSCOPE.
023356 112711 000377                MOVB  #377,(R1)
023362 022710 000377                CMP   #377,(R0)
023366 001403                       BEQ   EE3
023370 004737 023776          EE2:  JSR   PC,EEERR3
023374 000403                       BR    EE4
023376 005760 000002          EE3:  TST   2(R0)
023402 001372                       BNE   EE2

023404 062701 000001          EE4:  ADD   #1,R1
023410 005010                       CLR   (R0)
023412 005060 000002                CLR   2(R0)
023416 000240                       NOP                    ;FOR SCOPING WITH AN OSCILLOSCOPE.
023420 112711 000377                MOVB  #377,(R1)
023424 022710 177400                CMP   #177400,(R0)
023430 001403                       BEQ   EE6
023432 004737 023776          EE5:  JSR   PC,EEERR3
023436 000403                       BR    EE7
023440 005760 000002          EE6:  TST   2(R0)
023444 001372                       BNE   EE5

023446 062701 000001          EE7:  ADD   #1,R1
023452 005010                       CLR   (R0)
023454 005060 000002                CLR   2(R0)
023460 000240                       NOP                    ;FOR SCOPING WITH AN OSCILLOSCOPE.
023462 112711 000377                MOVB  #377,(R1)
023466 022760 000377 000002         CMP   #377,2(R0)
023474 001403                       BEQ   EE9
023476 004737 023776          EE8:  JSR   PC,EEERR3
023502 000402                       BR    EE10
023504 005710                 EE9:  TST   (R0)
023506 001373                       BNE   EE8

```

```
023510  062701  000001          EE10.   ADD     #1,R1
023514  005010                          CLR     (R0)
023516  005060  000002                  CLR     2(R0)
023522  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
023524  112711  000377                  MOVB    #377,(R1)
023530  022760  177400  000002          CMP     #177400,2(R0)
023536  001403                          BEQ     EE12
023540  004737  023776          EE11:   JSR     PC,EEERR3
023544  000402                          BR      EE13
023546  005710                  EE12:   TST     (R0)
023550  001373                          BNE     EE11

023552  000137  024056          EE13:   JMP     EEDONE

023556  000000                  EETMP1: .WORD   0
023560  000000                  EETMP2: .WORD   0
023562  000000                          .WORD   0


023564  032737  000002  177744  EEERR1: BIT     #2,@#MEMERR     ;SHOULD BE A MAIN MEMORY
023572  001002                          BNE     1$              ;ADDRESS AND CONTROL LINE
023574  000137  055440                  JMP     SPUR            ;PARITY ERROR.
023600  020137  177740          1$:     CMP     R1,@#LOADRS     ;ERROR ADDRESS SHOULD BE
023604  001402                          BEQ     EEERR2          ;TEST ADDRESS.
023606  000137  055440                  JMP     SPUR
023612  012637  001646          EEERR2: MOV     (SP)+,$TMP6
023616  005037  001670                  CLR     $TMP17
023622  005726                          TST     (SP)+           ;RESET THE STACK
023624  012737  000045  001672          MOV     #45,$TMP20
023632  013737  177740  001640          MOV     @#LOADRS,$TMP3
023640  013737  177742  001642          MOV     @#HIADRS,$TMP4
023646  013737  177744  001644          MOV     @#MEMERR,$TMP5
023654  010037  001646                  MOV     R0,$TMP6
023660  005037  001650                  CLR     $TMP7
023664  010037  001662                  MOV     R0,$TMP14
023670  062737  000002  001662          ADD     #2,$TMP14
023676  005037  001664                  CLR     $TMP15
023702  011037  001652                  MOV     (R0),$TMP10
023706  016037  000002  001654          MOV     2(R0),$TMP11
023714  010137  001656                  MOV     R1,$TMP12
023720  005037  001660                  CLR     $TMP13
023724  104045                  64$:    ERROR   45
023726  012737  177777  177744          MOV     #-1,@#MEMERR

023734  010002                          MOV     R0,R2
023736  020102                          CMP     R1,R2
023740  001002                          BNE     2$
023742  000137  023404                  JMP     EE4
023746  005202                  2$:     INC     R2
023750  020102                          CMP     R1,R2
023752  001002                          BNE     3$
023754  000137  023446                  JMP     EE7
023760  005202                  3$:     INC     R2
023762  020102                          CMP     R1,R2
023764  001002                          BNE     4$
023766  000137  023510                  JMP     EE10
```

```
023772  000137  024056          4$:     JMP     EEDONE


023776  011637  001656          EEERR3: MOV     (SP),$TMP12     ;REPORT FAILURE TO WRITE
                                                                ;THE CORRECT BYTE
024002  010037  001636                  MOV     R0,$TMP2
024006  005037  001640                  CLR     $TMP3
024012  010037  001642                  MOV     R0,$TMP4
024016  062737  000002  001642          ADD     #2,$TMP4
024024  005037  001644                  CLR     $TMP5
024030  011037  001646                  MOV     (R0),$TMP6
024034  016037  000002  001650          MOV     2(R0),$TMP7
024042  010137  001652                  MOV     R1,$TMP10
024046  005037  001654                  CLR     $TMP11
024052  104047                          ERROR   47
024054  000207                          RTS     PC


024056  104414                  EEDONE: RSET                    ;DONE.

                                ;;*************************************************************************
                                ;*TEST 23        CACHE ADDRESS MEMORY BYTE MASK GENERATOR, UNIBUS DATOB ZEROES TEST
                                ;*
                                ;*THIS IS ANOTHER TEST OF THE BYTE MASK GENERATION LIGIC.
                                ;*HERE UNIBUS DATOB'S WILL MOVE ZEROES INTO A BACKROUND
                                ;*PATTERN OF ONES.
                                ;*
                                ;;*************************************************************************
024060  000004                  TST23:  SCOPE
024062  012737  000010  001702          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
        000023                  HH=$TN-1
                                                                ;SET THE SKAD REGISTER
024070  012737  024760  055572          MOV     #TST24,SKAD     ;IN CASE THE TEST ABORTS.

024076  113737  001502  001632          MOVB    $TSTNM,$TMP0

024104  104416                          MMSKIP

024106  012737  024464  000114          MOV     #HHERR1,@#CACHVEC


024114  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                ;TO RELOCATE EVERYTHING
024120  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
024124  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
024130  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
024132  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                                ;RELOCATED TO THEMSELVES.
024136  012722  077406          64$:    MOV     #77406,(R2)+
024142  010401                          MOV     R4,R1
024144  072127  000006                  ASH     #6,R1
024150  010125                          MOV     R1,(R5)+
024152  005025                          CLR     (R5)+
024154  010410                          MOV     R4,(R0)
024156  062720  170000                  ADD     #170000,(R0)+
024162  062704  000200                  ADD     #200,R4
```

```
024166  077315                            SOB     R3,64$
024170  012710  177600                    MOV     #177600,(R0)
024174  012712  077406                    MOV     #77406,(R2)

024200  012737  000060  172516            MOV     #60,@#MMR3        ;TURN ON MEMORY MANAGEMENT
024206  012737  000001  177572            MOV     #1,@#MMR0         ;AND MAPPING BOX RELOCATION.

024214  012737  000014  177746            MOV     #MOM1,@#CONTRL    ;FORCE MISSES.

024222  012700  024460                    MOV     #HHTMP2,R0        ;LOCATE THE TEST SPACE.
024226  042700  000003                    BIC     #3,R0
024232  010001                            MOV     R0,R1

024234  012710  177777          HH1:      MOV     #-1,(R0)
024240  012760  177777  000002            MOV     #-1,2(R0)
024246  000240                            NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024250  105011                            CLRB    (R1)
024252  022710  177400                    CMP     #177400,(R0)
024256  001403                            BEQ     HH3
024260  004737  024676          HH2:      JSR     PC,HHERR3
024264  000404                            BR      HH4
024266  022760  177777  000002  HH3:      CMP     #-1,2(R0)
024274  001371                            BNE     HH2

024276  005201                  HH4:      INC     R1
024300  012710  177777                    MOV     #-1,(R0)
024304  012760  177777  000002            MOV     #-1,2(R0)
024312  000240                            NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024314  105011                            CLRB    (R1)
024316  022710  000377                    CMP     #377,(R0)
024322  001403                            BEQ     HH6
024324  004737  024676          HH5:      JSR     PC,HHERR3
024330  000404                            BR      HH7
024332  022760  177777  000002  HH6:      CMP     #-1,2(R0)
024340  001371                            BNE     HH5

024342  005201                  HH7:      INC     R1
024344  012710  177777                    MOV     #-1,(R0)
024350  012760  177777  000002            MOV     #-1,2(R0)
024356  000240                            NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024360  105011                            CLRB    (R1)
024362  122760  177400  000002            CMPB    #177400,2(R0)
024370  001403                            BEQ     HH9
024372  004737  024676          HH8:      JSR     PC,HHERR3
024376  000403                            BR      HH10
024400  022710  177777          HH9:      CMP     #-1,(R0)
024404  001372                            BNE     HH8

024406  005201                  HH10:     INC     R1
024410  012710  177777                    MOV     #-1,(R0)
024414  012760  177777  000002            MOV     #-1,2(R0)
024422  000240                            NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE.
024424  105011                            CLRB    (R1)
024426  022760  000377  000002            CMP     #377,2(R0)
024434  001403                            BEQ     HH12
024436  004737  024676          HH11:     JSR     PC,HHERR3
```

```
4773 024442 000403                              BR      HH13
4774 024444 022710 177777               HH12:   CMP     #-1,(R0)
4775 024450 001372                              BNE     HH11
4776
4777 024452 000137 024756               HH13:   JMP     HHDONE
4778
4779 024456 000000                      HHTMP1: .WORD   0
4780 024460 000000                      HHTMP2: .WORD   0                       ;THE TEST AREA
4781 024462 000000                              .WORD   0
4782
4783
4784 024464 032737 000002 177744        HHERR1: BIT     #2,@#MEMERR             ;SHOULD BE A MAIN MEMORY
4785 024472 001002                              BNE     1$                      ;ADDRESS AND CONTROL LINE
4786 024474 000137 055440                       JMP     SPUR                    ;PARITY ERROR.
4787 024500 020137 177740               1$:     CMP     R1,@#LOADRS             ;ERROR ADDRESS SHOULD BE
4788 024504 001402                              BEQ     HHERR2                  ;TEST ADDRESS.
4789 024506 000137 055440                       JMP     SPUR
4790 024512 012637 001646               HHERR2: MOV     (SP)+,$TMP6
4791 024516 005037 001670                       CLR     $TMP17
4792 024522 005726                              TST     (SP)+                   ;RESET THE STACK
4793 024524 012737 000051 001672                MOV     #51,$TMP20
4794 024532 013737 177740 001640                MOV     @#LOADRS,$TMP3
4795 024540 013737 177742 001642                MOV     @#HIADRS,$TMP4
4796 024546 013737 177744 001644                MOV     @#MEMERR,$TMP5
4797 024554 010037 001646                       MOV     R0,$TMP6
4798 024560 005037 001650                       CLR     $TMP7
4799 024564 010037 001662                       MOV     R0,$TMP14
4800 024570 062737 000002 001662                ADD     #2,$TMP14
4801 024576 005037 001664                       CLR     $TMP15
4802 024602 011037 001652                       MOV     (R0),$TMP10
4803 024606 016037 000002 001654                MOV     2(R0),$TMP11
4804 024614 010137 001656                       MOV     R1,$TMP12
4805 024620 005037 001660                       CLR     $TMP13
4806 024624 104051                      64$:    ERROR   51
4807 024626 012737 177777 177744                MOV     #-1,@#MEMERR
4808
4809 024634 010002                              MOV     R0,R2
4810 024636 020102                              CMP     R1,R2
4811 024640 001002                              BNE     2$
4812 024642 000137 024276                       JMP     HH4
4813 024646 005202                      2$:     INC     R2
4814 024650 020102                              CMP     R1,R2
4815 024652 001002                              BNE     3$
4816 024654 000137 024342                       JMP     HH7
4817 024660 005202                      3$:     INC     R2
4818 024662 020102                              CMP     R1,R2
4819 024664 001002                              BNE     4$
4820 024666 000137 024406                       JMP     HH10
4821 024672 000137 024756               4$:     JMP     HHDONE
4822
4823
4824 024676 011637 001656               HHERR3: MOV     (SP),$TMP12             ;REPORT FAILURE TO WRITE
4825                                                                            ;THE CORRECT BYTE
4826 024702 010037 001636                       MOV     R0,$TMP2
4827 024706 005037 001640                       CLR     $TMP3
4828 024712 010037 001642                       MOV     R0,$TMP4
```

```
024716  062737  000002  001642                  ADD     #2,$TMP4
024724  005037  001644                          CLR     $TMP5
024730  011037  001646                          MOV     (R0),$TMP6
024734  016037  000002  001650                  MOV     2(R0),$TMP7
024742  010137  001652                          MOV     R1,$TMP10
024746  005037  001654                          CLR     $TMP11
024752  104053                                  ERROR   53
024754  000207                                  RTS     PC


024756  104414                          HHDONE: RSET                            ;DONE!


                                        ;;*****************************************************************
                                        ;*TEST 24        CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST
                                        ;*
                                        ;*THIS TEST IS EXECUTED OPTIONALLY, ON THE CONDITION THAT
                                        ;*BIT 12 OF THE SWITCH REGISTER IS ON WHEN PROGRAM CONTROL
                                        ;*REACHES THIS POINT.  IF THIS SWITCH IS OFF THEN CONTROL
                                        ;*IS PASSED TO THE NEXT TEST.  THIS IS DONE BECAUSE THIS
                                        ;*TEST REQUIRES OPERATOR INTERVENTION.  THE USER IS ASKED TO
                                        ;*GO THROUGH A POWER DOWN-POWER UP SEQUENCE.  THEN
                                        ;*A SIMPLE SCAN IS MADE OF MEMORY WHICH CAUSES ALL
                                        ;*DATA AND ADDRESS MEMORY LOCATIONS IN THE CACHE TO BE
                                        ;*PARITY CHECKED.  IF THE POWER UP-CACHE INVLIDATER LOGIC
                                        ;*WORKED NO PARITY ERRORS CAN OCCUR.  BUT IF THIS INVALIDATER
                                        ;*FAILED THERE IS AN EXTREMELY HIGH PROBABILITY FOR THE
                                        ;*OCCURENCE OF A CACHE DATA OR CACHE ADDRESS PARITY ERROR.
                                        ;*IN FACT IF THE INVALIDATER CIRCUIT IS COMPLETELY INOPERATIVE
                                        ;*IT WILL BE VIRTUALLY IMPOSSIBLE TO RESTART THE PROGRAM.
                                        ;*WHEREAS MINOR OR NO FAILURES CAN AND WILL BE REPORTED.
                                        ;*IF NO PARITY ERRORS ARE ENCOUNTERED THE USER WILL
                                        ;*BE NOTIFIED SO THAT HE CAN KNOW IF A FATAL FAILURE
                                        ;*HAS OCCURRED.
                                        ;*
                                        ;;*****************************************************************
024760  000004                          TST24:  SCOPE
        000024                          DD=$TN-1
                                                                                ;SET THE SKAD REGISTER
024762  012737  025214  055572                  MOV     #TST25,SKAD             ;IN CASE THE TEST ABORTS.

024770  113737  001502  001632                  MOVB    $TSTNM,$TMP0
024776  012737  055440  000114                  MOV     #SPUR,@#CACHVEC ;INITIALLY EXPECT NO ERRORS.

025004  032777  010000  154526                  BIT     #SW12,@SWR              ;SEE IF THE USER HAS CHOSEN
025012  001002                                  BNE     DD1                     ;TO RUN THIS TEST, SW12=1.
025014  000177  030552                          JMP     @SKAD                   ;NO, SO GO TO NEXT TEST.

025020  012737  025152  000114          DD1:    MOV     #DDPER,@#CACHVEC                ;YES, SO SET UP THE PARITY
                                                                                ;ERROR VECTOR.
025026  013737  000024  025202                  MOV     @#24,DDTMP              ;SAVE THE OLD CONTENTS
025034  012737  025054  000024                  MOV     #DDPD,@#24              ;OF THE PWER FAIL TRAP
025042  005037  025204                          CLR     DDCNTR                  ;VECTOR AND RESET THIS
                                                                                ;VECTOR. CLEAR AN ERROR COUNT.
025046  104401                                  TYPE                            ;TELL THE USER TO POWER
025050  067140                                  .WORD   PDMSG1                  ;DOWN.
```

```
025052  000777                            BR      .                 ;WAIT, SHOULD THIS
                                                                    ;WAIT TIME OUT????
025054  000240                  DDPD:     NOP                       ;FOR SCOPE SYNC!
025056  012737  025066  000024            MOV     #DDPV,@#24        ;POWER DOWN ROUTINE
025064  000777                            BR      .                 ;JUST SET UP FOR POWER UP.
025066  012706  001400          DDPV:     MOV     #STACK,SP         ;RESET THE STACK POINTER
025072  013737  025202  000024            MOV     DDTMP,@#24        ;RESET POWER FAIL VECTOR.
025100  005000                            CLR     R0                ;SET UP FOR SCAN.
025102  012701  001000                    MOV     #1000,R1
025106  005720                  1$:       TST     (R0)+
025110  077102                            SOB     R1,1$
025112  013737  025202  000024  DDPU1:    MOV     DDTMP,@#24        ;RESET THE POWER FAIL VECTOR.
025120  005737  025204                    TST     DDCNTR            ;WERE THERE ANY ERRORS?
025124  001004                            BNE     DDPU2
025126  104401                            TYPE                      ;NO
025130  067316                            .WORD   PDMSG2
025132  000137  025206                    JMP     DDDONE

025136                          DDPU2:                              ;REPORT ERROR SUMMARY
025136  013737  025204  001636            MOV     DDCNTR,$TMP2
025144  104054                  1$:       ERROR   54
025146  000137  025206                    JMP     DDDONE

025152  032737  000360  177744  DDPER:    BIT     #360,@#MEMERR     ;THE ERROR SHOULD BE
025160  001406                            BEQ     DDPER1            ;A CACHE ADDRESS OR CACHE
025162  012737  177777  177744            MOV     #-1,MEMERR        ;DATA PARITY ERROR
025170  005237  025204                    INC     DDCNTR
025174  000002                            RTI

025176  000137  055440          DDPER1:   JMP     SPUR

025202  000000                  DDTMP:    .WORD   0                 ;STORAGE FOR POWER FAIL
                                                                    ;VECTORS OLD PC
025204  000000                  DDCNTR:   .WORD   0                 ;ERROR COUNT.

025206  104414                  DDDONE:   RSET
025210  012706  001400                    MOV     #STACK,SP

                                ;;*****************************************************************
                                ;*TEST 25         CACHE DATA MULTIPLEXER, CDMX, TEST
                                ;*
                                ;*THIS TEST PUTS DIFFERENT PATTERNS OF DATA AT THE INPUTS
                                ;*OF THE CDMX AND TESTS FOR PROPER SELECTION AND GOOD DATA.
                                ;*
                                ;;*****************************************************************
025214  000004                  TST25:    SCOPE
025216  012737  000010  001702            MOV     #10,$TIMES        ;;DO 10 ITERATIONS
                                                                    ;SET THE SKAD REGISTER
025224  012737  026322  055572            MOV     #TST26,SKAD       ;IN CASE THE TEST ABORTS.

025232  012737  055440  000114            MOV     #SPUR,@#CACHVEC ;PREPARE FOR UNEXPECTED ERRORS.
025240  113737  001502  001632            MOVB    $TSTNM,$TMP0
025246  012705  000006                    MOV     #6,R5             ;INITIALIZE
025252  012737  000004  026274            MOV     #4,JJCNT
025260  012700  026312                    MOV     #JJTMP2,R0
025264  042700  176002                    BIC     #176002,R0
```

```
025270  012701  140000                  MOV     #TESTR1,R1
025274  060001                          ADD     R0,R1
025276  012702  142000                  MOV     #TESTR2,R2
025302  060002                          ADD     R0,R2
025304  012703  144000                  MOV     #TESTR3,R3
025310  060003                          ADD     R0,R3
025312  012704  026300                  MOV     #JJPAT2,R4

025316  012737  125252  026276          MOV     #125252,JJPAT1  ;JJPAT1 CONTAINS THE DATA
                                                                ;WHICH WILL ENTER THE
                                                                ;MAIN MEMORY EVEN INPUTS
                                                                ;TO THE CDMX. INITIALLY
                                                                ;THIS WILL BE 125252
025324  012737  052525  026300          MOV     #52525,JJPAT2   ;DATA FOR MAIN MEMORY ODD
                                                                ;WORD INPUT TO CDMX
025332  005037  026302                  CLR     JJPAT3          ;GROUP 0 DATA INPUTS TO CDMX.
025336  012737  177777  026304          MOV     #-1,JJPAT4      ;GROUP 1 DATA INPUTS TO CDMX.
025344  012737  025344  001510  JJ1:    MOV     #JJ1,$LPERR
025352  013713  026276                  MOV     JJPAT1,(R3)     ;WRITE THE MAIN MEMORY
025356  013763  026300  000002          MOV     JJPAT2,2(R3)    ;EVEN AND ODD WORD PATTERNS

025364  012737  000034  177746          MOV     #S0M0M1,@#CONTRL        ;WRITE THE GROUP ZERO
025372  013711  026302                  MOV     JJPAT3,(R1)     ;PATTERN
025376  013761  026302  177776          MOV     JJPAT3,-2(R1)
025404  013761  026302  000002          MOV     JJPAT3,2(R1)
025412  005711                          TST     (R1)
025414  012737  000054  177746          MOV     #S1M0M1,@#CONTRL        ;WRITE THE GROUP ONE PATTERN
025422  013712  026304                  MOV     JJPAT4,(R2)
025426  013762  026304  177776          MOV     JJPAT4,-2(R2)
025434  013762  026304  000002          MOV     JJPAT4,2(R2)
025442  005712                          TST     (R2)

025444  005037  177746                  CLR     @#CONTRL
025450  000240                          NOP
025452                          JJ2:
025452  000240                          NOP
025454  016100  000000                  MOV     0(R1),R0
025460  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
025466  001011                          BNE     JJ3
025470  012737  000000  001634          MOV     #0,$TMP1
025476  010137  001636                  MOV     R1,$TMP2
025502  062737  000000  001636          ADD     #0,$TMP2
025510  104001                  66$:    ERROR   1
025512  020037  026302          JJ3:    CMP     R0,JJPAT3
025516  001406                          BEQ     65$
025520  012737  025532  001634          MOV     #64$,$TMP1
025526  010037  001636                  MOV     R0,$TMP2
025532  104005                  64$:    ERROR   5
025534                          65$:
025534  012737  025542  001510          MOV     #JJ4,$LPERR
025542                          JJ4:
025542  000240                          NOP
025544  016100  000002                  MOV     2(R1),R0
025550  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT.
025556  001011                          BNE     JJ5
025560  012737  000000  001634          MOV     #0,$TMP1
```

```
025566  010137  001636                  MOV     R1,$TMP2
025572  062737  000002  001636          ADD     #2,$TMP2
025600  104001                  66$:    ERROR   1
025602  020037  026302          JJ5:    CMP     R0,JJPAT3
025606  001406                          BEQ     65$
025610  012737  025622  001634          MOV     #64$,$TMP1
025616  010037  001636                  MOV     R0,$TMP2
025622  104005                  64$:    ERROR   5
025624                          65$:
025624  012737  025632  001510          MOV     #JJ6,$LPERR
025632                          JJ6:
025632  000240                          NOP
025634  016200  000000                  MOV     0(R2),R0
025640  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT.
025646  001011                          BNE     JJ7
025650  012737  000001  001634          MOV     #1,$TMP1
025656  010237  001636                  MOV     R2,$TMP2
025662  062737  000000  001636          ADD     #0,$TMP2
025670  104001                  66$:    ERROR   1
025672  020037  026304          JJ7:    CMP     R0,JJPAT4
025676  001406                          BEQ     65$
025700  012737  025712  001634          MOV     #64$,$TMP1
025706  010037  001636                  MOV     R0,$TMP2
025712  104006                  64$:    ERROR   6
025714                          65$:
025714  012737  025722  001510          MOV     #JJ8,$LPERR
025722                          JJ8:
025722  000240                          NOP
025724  016200  000002                  MOV     2(R2),R0
025730  032737  000010  177752          BIT     #10,@#HITMIS    ;MUST BE A HIT!
025736  001011                          BNE     JJ9
025740  012737  000001  001634          MOV     #1,$TMP1
025746  010237  001636                  MOV     R2,$TMP2
025752  062737  000002  001636          ADD     #2,$TMP2
025760  104001                  66$:    ERROR   1
025762  020037  026304          JJ9:    CMP     R0,JJPAT4
025766  001406                          BEQ     65$
025770  012737  026002  001634          MOV     #64$,$TMP1
025776  010037  001636                  MOV     R0,$TMP2
026002  104006                  64$:    ERROR   6
026004                          65$:
026004  012737  026012  001510          MOV     #JJ10,$LPERR
026012  000240                  JJ10:   NOP
026014  012737  000014  177746          MOV     #M1M0,@#CONTRL  ;CHECK MAIN MEMORY DATA
026022  011300                          MOV     (R3),R0         ;EVEN WORD
026024  020037  026276                  CMP     R0,JJPAT1
026030  001403                          BEQ     1$
026032  010037  001636                  MOV     R0,$TMP2
026036  104007                          ERROR   7
026040  012737  026046  001510  1$:     MOV     #JJ11,$LPERR
026046  016300  000002          JJ11:   MOV     2(R3),R0        ;CHECK MAIN MEMORY EVEN
026052  020037  026300                  CMP     R0,JJPAT2       ;WORD
026056  001403                          BEQ     JJ12
026060  010037  001636                  MOV     R0,$TMP2
026064  104010                  1$:     ERROR   10

```

```
026066  005037  177746          JJ12:    CLR     @#CONTRL
026072  020427  026304                   CMP     R4,#JJPAT4          ;NOW GET EVERY PERMUTATION
026076  001011                           BNE     JJ13                ;OF THE FOUR TEST PATTERNS:
                                                                     ;125252,052525,177777 AND
026100  011437  026306                   MOV     (R4),JJPAT5         ;000000 INTO JJPAT1, JJPAT2,
026104  013714  026300                   MOV     JJPAT2,(R4)         ;JJPAT3 AND JJPAT4 AND
026110  012704  026300                   MOV     #JJPAT2,R4          ;REPEAT THE TEST.
026114  013714  026306                   MOV     JJPAT5,(R4)
026120  000406                           BR      JJ14

026122  012437  026306          JJ13:    MOV     (R4)+,JJPAT5
026126  011464  177776                   MOV     (R4),-2(R4)
026132  013714  026306                   MOV     JJPAT5,(R4)

026136  005305                  JJ14:    DEC     R5
026140  001402                           BEQ     1$
026142  000137  025344                   JMP     JJ1
026146  012705  000006          1$:      MOV     #6,R5
026152  013737  026276  026306           MOV     JJPAT1,JJPAT5
026160  005337  026274                   DEC     JJCNT

026164  023727  026274  000003           CMP     JJCNT,#3
026172  001010                           BNE     JJ15
026174  013737  026300  026276           MOV     JJPAT2,JJPAT1
026202  013737  026306  026300           MOV     JJPAT5,JJPAT2
026210  000137  025344                   JMP     JJ1

026214  023727  026274  000002  JJ15:    CMP     JJCNT,#2
026222  001010                           BNE     JJ16
026224  013737  026302  026276           MOV     JJPAT3,JJPAT1
026232  013737  026306  026302           MOV     JJPAT5,JJPAT3
026240  000137  025344                   JMP     JJ1

026244  023727  026274  000001  JJ16:    CMP     JJCNT,#1
026252  001023                           BNE     JJ17                ;DONE?
026254  013737  026304  026276           MOV     JJPAT4,JJPAT1
026262  013737  026306  026304           MOV     JJPAT5,JJPAT4
026270  000137  025344                   JMP     JJ1

026274  000000                  JJCNT:   .WORD   0                   ;COUNTER USED TO GENERATE
                                                                     ;PERMUTATIONS.
026276  000000                  JJPAT1:  .WORD   0                   ;MAIN MEMORY EVEN WORD DATA PATTERN
026300  000000                  JJPAT2:  .WORD   0                   ;MAIN MEMORY ODD WORD DATA PATTERN
026302  000000                  JJPAT3:  .WORD   0                   ;GROUP 0 DATA PATTERN
026304  000000                  JJPAT4:  .WORD   0                   ;GROUP 1 DATA PATTERN
026306  000000                  JJPAT5:  .WORD   0                   ;TEMPORARY STORAGE

026310  000000                  JJTMP1:  .WORD   0                   ;TEST AREA, SO CODE WON'T
026312  000000  000000  000000  JJTMP2:  .WORD   0,0,0,0             ;OVER LAP THE HITS OF
026320  000000
                                                                     ;THE TEST WORDS.

026322                          JJ17:                                ;DONE.

                                ;;*****************************************************************
                                ;*TEST 26        CACHE DATA MEMORY ADDRESS DRIVERS TEST
```

```
                                        ;*
                                        ;*THIS TEST PERFORMS A DUAL ADDRESS TEST ON THE
                                        ;*CACHE DATA MEMORIES OF BOTH GROUPS.
                                        ;*
                                        ;;*************************************************************
026322  000004                          TST26:  SCOPE
026324  012737  000010  001702                  MOV     #10,$TIMES      ;;DO 10 ITERATIONS
                                                                        ;SET THE SKAD REGISTER
026332  012737  027032  055572                  MOV     #TST27,SKAD     ;IN CASE THE TEST ABORTS.

026340  012737  055440  000114                  MOV     #SPUR,@#CACHVEC
026346  113737  001502  001632                  MOVB    $TSTNM,$TMP0

026354  012737  000001  027024          GG1:    MOV     #1,GGFLG1       ;INITIALIZE FOR A TEST
026362  012737  000054  027026                  MOV     #S1MOM1,GGGS    ;ON GROUP 1 FIRST
026370  012737  000034  027030                  MOV     #SOMOM1,GGGM    ;SOM1 AND S1MO ARE PATTERNS
                                                                        ;DESTINED FOR THE CACHE
                                                                        ;CONTROL REGISTER
026376  012700  026376                  GG2:    MOV     #GG2,R0         ;MAKE THIS CODE, LOCATIONS
026402  012701  001000                          MOV     #1000,R1        ;GG2 THROUGH GG2+2000(OCT),
026406  013737  027026  177746          GG3:    MOV     GGGS,@#CONTRL   ;HITS IN THE GROUP NOT
026414  005760  002000                          TST     2000(R0)        ;BEING TESTED AND MISSES
026420  013737  027030  177746                  MOV     GGGM,@#CONTRL   ;IN THE GROUP BEING TESTED.
026426  005720                                  TST     (R0)+
026430  077112                                  SOB     R1,GG3
026432  013700  027026                          MOV     GGGS,R0         ;MAKE THE TEST AREA
026436  042700  177717                          BIC     #177717,R0      ;HITS IN THE GROUP
026442  010037  177746                          MOV     R0,@#CONTRL     ;BEING TESTED
026446  012701  140000                          MOV     #TESTR1,R1
026452  012700  001000                          MOV     #1000,R0
026456  012737  026464  001510                  MOV     #GG4,$LPERR
026464  000240                          GG4:    NOP
026466  005011                                  CLR     (R1)
026470  005711                                  TST     (R1)
026472  005711                                  TST     (R1)
026474  032737  000010  177752                  BIT     #10,@#HITMISS
026502  001006                                  BNE     2$
026504  013737  027024  001634                  MOV     GGFLG1,$TMP1
026512  010137  001636                          MOV     R1,$TMP2
026516  104001                          1$:     ERROR   1
026520  005721                          2$:     TST     (R1)+
026522  077020                                  SOB     R0,GG4
026524  013700  027030                          MOV     GGGM,R0         ;FROM HERE ON SELECT
026530  042700  177717                          BIC     #177717,R0      ;THE GROUP NOT BEING
026534  010037  177746                          MOV     R0,@#CONTRL     ;TESTED

026540  012701  140000                          MOV     #TESTR1,R1
026544  012700  001000                          MOV     #1000,R0
026550  012737  026556  001510                  MOV     #GG5,$LPERR
026556  000240                          GG5:    NOP                     ;
026560  010111                                  MOV     R1,(R1)         ;WRITE #ADDRESS INTO @#ADDRESS.
026562  005721                                  TST     (R1)+
026564  077004                                  SOB     R0,GG5

026566  012701  140000                          MOV     #TESTR1,R1
026572  012700  001000                          MOV     #1000,R0
```

```
026576  012737  026604  001510             MOV     #GG6,$LPERR
026604  000240                    GG6:     NOP
026606  011102                             MOV     (R1),R2          ;READ BACK THE ADDRESS
026610  032737  000010  177752             BIT     #10,@#HITMIS
026616  001006                             BNE     GG7
026620  013737  027024  001634             MOV     GGFLG1,$TMP1

026626  010137  001636                     MOV     R1,$TMP2
026632  104001                    1$:      ERROR   1

026634  020102                    GG7:     CMP     R1,R2            ;DOES @#ADDRESS CONTAIN
026636  001412                             BEQ     GG8              ;#ADDRESS

026640  013737  027024  001634             MOV     GGFLG1,$TMP1
026646  010137  001636                     MOV     R1,$TMP2
026652  010237  001640                     MOV     R2,$TMP3
026656  010137  001642                     MOV     R1,$TMP4
026662  104016                    1$:      ERROR   16

026664  005121                    GG8:     COM     (R1)+            ;COMPLIMENT DATA
026666  077032                             SOB     R0,GG6           ;LOOP FOR NEXT ADDRESS.
026670  012701  140000                     MOV     #TESTR1,R1
026674  012700  001000                     MOV     #1000,R0
026700  012737  026706  001510             MOV     #GG9,$LPERR
026706  000240                    GG9:     NOP
026710  011102                             MOV     (R1),R2          ;GO BACK AND CHECK
026712  032737  000010  177752             BIT     #10,@#HITMIS     ;COMPLIMENTED DATA
026720  001006                             BNE     GG10
026722  013737  027024  001634             MOV     GGFLG1,$TMP1
026730  010137  001636                     MOV     R1,$TMP2
026734  104001                    1$:      ERROR   1
                                                                    ;?????

026736  010103                    GG10:    MOV     R1,R3            ;IS COMPLIMENT DATA CORRECT?
026740  005103                             COM     R3
026742  020302                             CMP     R3,R2
026744  001412                             BEQ     GG11
026746  013737  027024  001634             MOV     GGFLG1,$TMP1
026754  010337  001636                     MOV     R3,$TMP2
026760  010237  001640                     MOV     R2,$TMP3
026764  010137  001642                     MOV     R1,$TMP4
026770  104016                    1$:      ERROR   16

026772  005721                    GG11:    TST     (R1)+            ;TEST NEXT LOCATION
026774  077034                             SOB     R0,GG9

026776  012737  000034  027026             MOV     #S0M0M1,GGGS     ;GO BACK AND RUN
027004  012737  000054  027030             MOV     #S1M0M1,GGGM     ;TEST IN GROUP 0.
027012  005337  027024                     DEC     GGFLG1
027016  001005                             BNE     GG12
027020  000137  026376                     JMP     GG2

027024  000000                    GGFLG1:  .WORD   0                ;GROUP BEING TESTED, 0 OR 1.

027026  000000                    GGGS:    .WORD   0                ;CACHE CONTROL REGISTER
027030  000000                    GGGM:    .WORD   0                ;PATTERNS
```

```

027032                              GG12:                           ;DONE

                                    ;;*************************************************************
                                    ;*TEST 27       CACHE DATA MEMORY COUNT PATTERN TEST
                                    ;*
                                    ;*THIS TEST RUNS A COUNT PATTERN THROUGH EACH LOCATION
                                    ;*OF THE CACHE DATA MEMORY FOR EACH GROUP.
                                    ;*
                                    ;;*************************************************************
027032  000004                      TST27:  SCOPE
027034  012737  000010  001702              MOV     #10,$TIMES      ;;DO 10 ITERATIONS
                                                                    ;SET THE SKAD REGISTER
027042  012737  030012  055572              MOV     #TST30,SKAD     ;IN CASE THE TEST ABORTS.

027050  012737  055440  000114              MOV     #SPUR,@#CACHVEC
027056  113737  001502  001632              MOVB    $TSTNM,$TMP0

027064  012737  000001  027520      LL1:    MOV     #1,LLFLG1       ;TEST GROUP ONE FIRST
027072  012737  000044  027526              MOV     #S1M0,LLGS      ;S1M0 AND S0M1 ARE PATTERNS
027100  012737  000030  027530              MOV     #S0M1,LLGM      ;WHICH WILL BE LOADED INTO
027106  012737  027106  001510      LL2:    MOV     #LL2,$LPERR     ;THE CACHE CONTROL REGISTER.
027114  012737  055440  000114              MOV     #SPUR,@#CACHVEC
027122  012700  027106                      MOV     #LL2,R0         ;MAKE THIS CODE, LOCATIONS
027126  012701  001000                      MOV     #1000,R1        ;LL2 THROUGH LL2+2000 (OCT)
                                                                    ;HITS IN THE CACHE GROUP
027132  013737  027530  177746      LL3:    MOV     LLGM,@#CONTRL   ;NOT BEING TESTED, AND MISSES
027140  005710                              TST     (R0)            ;TO THE CACHE GROUP BEING
027142  013737  027526  177746              MOV     LLGS,@#CONTRL   ;TESTED.
027150  005760  002000                      TST     2000(R0)
027154  062700  000002                      ADD     #2,R0
027160  077114                              SOB     R1,LL3

027162  012701  140000                      MOV     #TESTR1,R1      ;MAKE THE MEMORY TEST AREA
027166  012700  001000                      MOV     #1000,R0        ;HITS IN THE GROUP BEING
027172  012737  027214  001510              MOV     #1$,$LPERR      ;TESTED.
027200  013702  027526                      MOV     LLGS,R2
027204  042702  177717                      BIC     #177717,R2
027210  010237  177746                      MOV     R2,@#CONTRL
027214  005011                      1$:     CLR     (R1)
027216  005711                              TST     (R1)
027220  005721                              TST     (R1)+
027222  032737  000010  177752              BIT     #10,@#HITMIS
027230  001011                              BNE     3$
027232  013737  027520  001634              MOV     LLFLG1,$TMP1
027240  011137  001636                      MOV     (R1),$TMP2
027244  062737  177776  001636              ADD     #-2,$TMP2
027252  104001                      2$:     ERROR   1
027254  077021                      3$:     SOB     R0,1$
027256  013700  027530                      MOV     LLGM,R0         ;FROM NOW ON SELECT
027262  042700  177717                      BIC     #177717,R0      ;THE GROUP NOT BEING
027266  010037  177746                      MOV     R0,@#CONTRL     ;TESTED

027272  012701  140000                      MOV     #TESTR1,R1      ;INITIALIZE FOR TEST.
027276  012700  001000                      MOV     #1000,R0        ;COUNTER.
027302  005002                      LL4:    CLR     R2              ;DATA PATTERN WRITTEN
```

```
027304  005003                           CLR     R3              ;LOGICAL 'OR' OF BAD DATA
027306  012704  177777                   MOV     #177777,R4      ;LOGICAL 'AND' OF BAD DATA
027312  005005                           CLR     R5              ;DATA PATTERN READ
027314  005037  027532                   CLR     LLCNT1          ;NUMBER OF LOCATIONS WHICH FAIL.
027320  005037  027522                   CLR     LLFLG2          ;ERROR IN GROUP FLAG
027324  012737  027332  001510           MOV     #LL5,$LPERR
027332  005037  027524          LL5:     CLR     LLFLG4          ;ERROR IN TESTED WORD FLAG.
027336  000240                           NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
027340  010211                           MOV     R2,(R1)
027342  011105                           MOV     (R1),R5
027344  032737  000010  177752           BIT     #10,@#HITMIS
027352  001006                           BNE     LL6
027354  013737  027520  001634           MOV     LLFLG1,$TMP1
027362  010137  001636                   MOV     R1,$TMP2
027366  104001                  1$:      ERROR   1
027370  020205                  LL6:     CMP     R2,R5           ;GOOD DATA
027372  001402                           BEQ     LL7
027374  000137  027744                   JMP     LLERR2          ;BAD DATA BUT NO TRAP OR
                                                                 ;ABORT OCCURRED!
027400                          LL7:                             ;DECREMENT THE COUNT PATTERN
                                                                 ;AND LOOP IF NOT DONE
027400  005737  027524                   TST     LLFLG4          ;IF THERE WAS AN ERROR
027404  001405                           BEQ     LL8             ;IN THE WORD JUST TESTED
027406  005237  027532                   INC     LLCNT1          ;INCREMENT LLCNT1
027412  012737  177777  027522           MOV     #-1,LLFLG2      ;AND SET ERROR IN GROUP FLAG.
027420  062701  000002          LL8:     ADD     #2,R1           ;GO TO NEXT WORD.
027424  077036                           SOB     R0,LL5

027426  005737  027522                   TST     LLFLG2          ;DONE WITH THAT GROUP,
027432  001417                           BEQ     LL9             ;SEE IF THERE WERE
027434  112737  000013  001514           MOVB    #13,$ITEMB      ;ANY ERRORS. IF SO THEN
027442  013737  027520  001634           MOV     LLFLG1,$TMP1    ;PRINT AN ERROR SUMMARY
027450  010437  001636                   MOV     R4,$TMP2        ;FOR THAT GROUP.
027454  010337  001640                   MOV     R3,$TMP3
027460  013737  027532  001642           MOV     LLCNT1,$TMP4
027466  004737  056354                   JSR     PC,ERTYPE

027472  012737  000044  027530  LL9:     MOV     #S1M0,LLGM      ;TEST THE OTHER GROUP, 0.
027500  012737  000030  027526           MOV     #S0M1,LLGS      ;
027506  005337  027520                   DEC     LLFLG1
027512  001137                           BNE     LL10            ;DONE?
027514  000137  027106                   JMP     LL2

027520  000000                  LLFLG1:  .WORD   0               ;GROUP BEING TESTED, 1 OR 0.
027522  000000                  LLFLG2:  .WORD   0               ;ERROR OCCURRED IN GROUP FLAG.

027524  000000                  LLFLG4:  .WORD   0               ;ERROR OCCURRED IN WORD FLAG.

027526  000000                  LLGS:    .WORD   0               ;PATTERNS FOR CONTROL REGISTER
027530  000000                  LLGM:    .WORD   0

027532  000000                  LLCNT1:  .WORD   0               ;GROUP ERROR COUNT

027534  000000                  LLMER:   .WORD   0               ;TEMPORARY STORAGE FOR
                                                                 ;THE CACHE ERROR REGISTER.
027536  000000                  LLTMP1:  .WORD   0
```

```

027540  013737  177744  027534  LLERR1: MOV     @#MEMERR,LLMER   ;COME HERE ON PARITY
027546  012737  004100  027536          MOV     #4100,LLTMP1     ;ABORT OR TRAP.
027554  005737  027520                  TST     LLFLG1           ;TESTING GROUP 1 OR 0?
027560  001403                          BEQ     1$
027562  012737  004200  027536          MOV     #4200,LLTMP1
027570  023737  027536  027534  1$:     CMP     LLTMP1,LLMER     ;WAS THE ERROR EXPECTED?
027576  001402                          BEQ     2$
027600  000137  055440                  JMP     SPUR             ;NO!

027604  020137  177740          2$:     CMP     R1,@#LOADRS      ;WAS THAT ADDRESS EXPECTED?
027610  001402                          BEQ     3$
027612  000137  055440                  JMP     SPUR             ;NO.

027616  012737  177777  027524  3$:     MOV     #-1,LLFLG4       ;SET WORD ERROR FLAG
027624  050203                          BIS     R2,R3            ;DO 'OR' OF FAILING DATA
027626  005102                          COM     R2
027630  040204                          BIC     R2,R4            ;DO 'AND' OF FAILING DATA
027632  005102                          COM     R2
027634  011637  001634                  MOV     (SP),$TMP1
027640  022626                          CMP     (SP)+,(SP)+
027642  013737  027520  001636          MOV     LLFLG1,$TMP2
027650  010237  001640                  MOV     R2,$TMP3
027654  010137  001650                  MOV     R1,$TMP7
027660  013737  177740  001642          MOV     @#LOADRS,$TMP4
027666  013737  177742  001644          MOV     @#HIADRS,$TMP5
027674  042737  140000  001644          BIC     #140000,$TMP5
027702  013737  027534  001646          MOV     LLMER,$TMP6
027710  104011                          ERROR   11               ;REPORT ERROR.

027712  012737  027724  000114          MOV     #LLERR3,@#CACHVEC        ;BEFORE CONTINUING THE
                                                                 ;BAD PARITY IN THE WORD
                                                                 ;BEING TESTED MUST BE
                                                                 ;DEALT WITH!
027720  005011                          CLR     (R1)             ;THIS INSTRUCTION CLR (R1)
027722  005711                          TST     (R1)             ;SHOULD TRAP!

027724  012737  177777  177744  LLERR3: MOV     #-1,@#MEMERR     ;CLR THE ERROR REGISTER
027732  012737  027540  000114          MOV     #LLERR1,@#CACHVEC        ;RESTORE THE PARITY ERROR
027740  000137  027400                  JMP     LL7              ;VECTOR AND CONTINUE.

027744  012737  177777  027524  LLERR2: MOV     #-1,LLFLG4       ;BAD DATA WAS READ BUT
                                                                 ;NO TRAP OR ABORT OCCURRED.
027752  050203                          BIS     R2,R3            ;'OR' BAD CATA
027754  005102                          COM     R2
027756  040204                          BIC     R2,R4            ;'AND' BAD DATA
027760  005102                          COM     R2
027762  013737  027520  001634          MOV     LLFLG1,$TMP1
027770  010137  001640                  MOV     R1,$TMP3
027774  010237  001642                  MOV     R2,$TMP4
030000  010537  001644                  MOV     R5,$TMP5

030004  104012                  1$:     ERROR   12               ;REPORT ERROR.

030006  000137  027400                  JMP     LL7              ;CONTINUE TEST.
030012                          LL10:
```

```


                                        ;;*************************************************************
                                        ;*TEST 30        CACHE DATA MEMORY PARITY CHECKERS LOW BYTE TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE TWO CACHE DATA MEMORY PARITY
                                        ;*CHECKERS FOR THE LOW BYTE, ONE FOR EACH GROUP. THE
                                        ;*MAINTENANCE REGISTER ISUSED TO FORCE A PARITY A
                                        ;*PARITY ERROR AT EVERY DATA PATTERN WHICH HAS A ONE
                                        ;*PARITY BIT. NOTE THAT THE CACHE DATA MEMORY PARITY HAS,
                                        ;*EFFECTIVELY, ODD PARITY. THE MAINTENANCE FUNCTION ON THE
                                        ;*CACHE DATA MEMORY PARITY CHECKERS HAS THE EFFECT OF
                                        ;*FORCING THE PARITY BIT OF THE BYTE BEING CHECKED TO
                                        ;*ZERO. THIS MEANS THAT ONCE THIS MAINTENANCE FUNCTION
                                        ;*IS ENABLED THE ERROR WILL OCCUR ON A SUBSEQUENT
                                        ;*READ OF A BYTE WITH A ONE PARITY BIT, THAT IS
                                        ;*BYTES WITH ZERO PARITY BITS WILL NOT CAUSE THE ERROR.
                                        ;*
                                        ;;*************************************************************
030012  000004                          TST30:  SCOPE
030014  012737  000020  001702                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000031                          IIA-$TN
                                                                        ;SET THE SKAD REGISTER
030022  012737  030464  055572                  MOV     #TST31,SKAD     ;IN CASE THE TEST ABORTS.

030030  113737  001502  001632                  MOVB    $TSTNM,$TMPO
030036  012737  055440  000114                  MOV     #SPUR,@#CACHVEC

030044  005000                                  CLR     R0              ;THIS IS THE COUNTER CONTAINING
                                                                        ;THE TEST DATA PATTERN
030046  012737  030046  001510          IIA1:   MOV     #IIA1,$LPERR
030054  004737  056032                          JSR     PC,PARCNT       ;SET IF THIS TEST PATTERN HAS
030060  032702  000001                          BIT     #BIT0,R2        ;THE PARITY BIT SET (1), IF NOT
030064  001402                                  BEQ     IIA2            ;GO TO THE NEXT PATTERN
030066  000137  030444                          JMP     IIA7
030072  012737  000030  177746          IIA2:   MOV     #SOM1,@#CONTRL  ;SELECT GROUP ZERO.
030100  012737  030350  000114                  MOV     #IIAR1,@#CACHVEC        ;SET UP FOR THE ERROR
030106  012705  030346                          MOV     #IIAT1,R5       ;MAKE THE TEST ADDRESS A
030112  005715                                  TST     (R5)            ;HIT IN GROUP ZERO
030114  005715                                  TST     (R5)            ;MAKE SURE IT IS A HIT

                                                                        ;SEE IF REFERENCE ADDRESS
030116  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
030124  001007                                  BNE     1$
                                                                        ;IF NOT ERROR!
030126  010537  001636                          MOV     R5,$TMP2
030132  012737  000000  001634                  MOV     #0,$TMP1
030140  104001                                  ERROR   1

030142  104415                                  SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.


030144  012704  000020                  1$:     MOV     #20,R4  ;THIS PATTERN WILL BE
030150  012702  177750                          MOV     #MAINT,R2       ;PUT IN THE MAINTENANCE
030154  005001                                  CLR     R1              ;REGISTER
030156  010015                                  MOV     R0,(R5)         ;PUT THE TEST PATTERN IN
```

```
                                                                ;THE TEST ADDRESS
030160  000401                                  BR      64$

        030162                                  LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY..
        030160                                  LOC=-4&LOC
        030164                                  LOC=LOC+4
        030164                                  .=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'
030164  010412                          64$:    MOV     R4,(R2)         ;TURN ON THE MAINT. REG.
030166  021500                                  CMP     (R5),R0         ;THE REFERENCE TO (R5)
030170  010112                                  MOV     R1,(R2)         ;SHOULD CAUSE THE ERROR.

030172                                  IIA3:
                                                                ;THE ERROR DIDN'T OCCUR.
030172  010037  001636                          MOV     R0,$TMP2        ;REPORT FAILURE
030176  012737  030346  001640                  MOV     #IIAT1,$TMP3
030204  005037  001642                          CLR     $TMP4
030210  104144                          64$:    ERROR   144

030212  012737  030410  000114          IIA4:   MOV     #IIAR2,@#CACHVEC        ;SET UP FOR THE GROUP ONE
030220  012737  030212  001510                  MOV     #IIA4,$LPERR    ;ERROR
030226  012737  000044  177746                  MOV     #S1M0,@#CONTRL  ;SELECT GROUP ONE

030234  012705  030346                          MOV     #IIAT1,R5       ;MAKE THE TEST ADDRESS A
030240  005715                                  TST     (R5)            ;HIT, IN GROUP ONE.
030242  005715                                  TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
030244  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
030252  001007                                  BNE     1$
                                                                ;IF NOT ERROR!
030254  010537  001636                          MOV     R5,$TMP2
030260  012737  000001  001634                  MOV     #1,$TMP1
030266  104001                                  ERROR   1

030270  104415                                  SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.


030272  012704  000100                  1$:     MOV     #100,R4 ;THIS PATTERN WILL BE
030276  012702  177750                          MOV     #MAINT,R2       ;PUT IN THE MAINT. REG.
030302  005001                                  CLR     R1
030304  010015                                  MOV     R0,(R5)         ;PUT THE TEST PATTERN IN (R5),
                                                                ;IIAT1.
030306  000402                                  BR      50$             ;PUT THE NEXT INSTRUCTION EXECUTED
                                                                ;ON AN EVEN WORD BOUNDARY SO THE
                                                                ;SUBSEQUENT INSTRUCTION, A CMP,
                                                                ;WILL BE A HIT.

        030310                                  LOC-.           ;GET THE PC TO AN EVEN WORD BOUNDARY...
        030310                                  LOC=-4&LOC
        030314                                  LOC=LOC+4
        030314                                  .=LOC
```

```

030314  000240                  50$:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE.
030316  010412                          MOV     R4,(R2)         ;TURN ON THE MAINT. REG.
030320  021500                          CMP     (R5),R0         ;THIS REFERENCE TO (R5) SHOULD
030322  010112                          MOV     R1,(R2)         ;CAUSE THE ERROR.

030324                          IIA5:
                                                                ;THE ERROR DIDN'T OCCUR!
030324  010037  001636                  MOV     R0,$TMP2        ;REPORT FAILURE
030330  012737  030346  001640          MOV     #IIAT1,$TMP3
030336  005037  001642                  CLR     $TMP4
030342  104145                  64$:    ERROR   145

030344  000437                  IIA6:   BR      IIA7

030346  000000                  IIAT1:.WORD     0

030350                          IIAR1:
030350  022737  004500  177744          CMP     #4500,@#MEMERR  ;MAKE SURE THE ERROR
030356  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
030360  000137  055440          1$:     JMP     SPUR
030364  022737  030346  177740  2$:     CMP     #IIAT1,@#LOADRS ;MAKE SURE THE ERROR
030372  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
030374  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
030376  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
030404  000137  030212                  JMP     IIA4            ;GO TEST GROUP ONE
030410                          IIAR2:
030410  022737  004600  177744          CMP     #4600,@#MEMERR  ;MAKE SURE THE ERROR
030416  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
030420  000137  055440          1$:     JMP     SPUR
030424  022737  030346  177740  2$:     CMP     #IIAT1,@#LOADRS ;MAKE SURE THE ERROR
030432  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
030434  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
030436  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.

030444  022700  000377          IIA7:   CMP     #377,R0         ;INCREMENT THE TEST
030450  001404                          BEQ     IIA8            ;PATTERN
030452  062700  000001                  ADD     #1,R0
030456  000137  030046                  JMP     IIA1

030462  104414                  IIA8:   RSET

                                ;;**************************************************************
                                ;*TEST 31       CACHE DATA MEMORY PARITY CHECKERS HIGH BYTE TEST
                                ;*
                                ;*THIS IS A TEST OF THE TWO CACHE DATA MEMORY PARITY
                                ;*CHECKERS FOR THE HIGH BYTE, ONE FOR EACH GROUP. THE
                                ;*MAINTENANCE REGISTER ISUSED TO FORCE A PARITY A
                                ;*PARITY ERROR AT EVERY DATA PATTERN WHICH HAS A ONE
                                ;*PARITY BIT. NOTE THAT THE CACHE DATA MEMORY PARITY HAS,
                                ;*EFFECTIVELY, ODD PARITY. THE MAINTENANCE FUNCTION ON THE
                                ;*CACHE DATA MEMORY PARITY CHECKERS HAS THE EFFECT OF
                                ;*FORCING THE PARITY BIT OF THE BYTE BEING CHECKED TO
                                ;*ZERO. THIS MEANS THAT ONCE THIS MAINTENANCE FUNCTION
```

```
                                        ;*IS ENABLED THE ERROR WILL OCCUR ON A SUBSEQUENT
                                        ;*READ OF A BYTE WITH A ONE PARITY BIT, THAT IS
                                        ;*BYTES WITH ZERO PARITY BITS WILL NOT CAUSE THE ERROR.
                                        ;*
                                        ;;*************************************************************
030464  000004                          TST31:  SCOPE
030466  012737  000020  001702                  MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000032                          IIB=$TN
                                                                        ;SET THE SKAD REGISTER
030474  012737  031140  055572                  MOV     #TST32,SKAD     ;IN CASE THE TEST ABORTS.

030502  113737  001502  001632                  MOVB    $TSTNM,$TMP0
030510  012737  055440  000114                  MOV     #SPUR,@#CACHVEC

030516  005000                                  CLR     R0              ;THIS IS THE COUNTER CONTAINING
                                                                        ;THE TEST DATA PATTERN
030520  012737  030520  001510          IIB1:   MOV     #IIB1,$LPERR
030526  004737  056032                          JSR     PC,PARCNT       ;SET IF THIS TEST PATTERN HAS
030532  032702  000001                          BIT     #BIT0,R2        ;THE PARITY BIT SET (1), IF NOT
030536  001402                                  BEQ     IIB2            ;GO TO THE NEXT PATTERN
030540  000137  031120                          JMP     IIB7
030544  012737  000030  177746          IIB2:   MOV     #S0M1,@#CONTRL  ;SELECT GROUP ZERO.
030552  012737  031024  000114                  MOV     #IIBR1,@#CACHVEC        ;SET UP FOR THE ERROR
030560  012705  031022                          MOV     #IIBT1,R5       ;MAKE THE TEST ADDRESS A
030564  005715                                  TST     (R5)            ;HIT IN GROUP ZERO
030566  005715                                  TST     (R5)            ;MAKE SURE IT IS A HIT

                                                                        ;SEE IF REFERENCE ADDRESS
030570  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
030576  001007                                  BNE     1$
                                                                        ;IF NOT ERROR!
030600  010537  001636                          MOV     R5,$TMP2
030604  012737  000000  001634                  MOV     #0,$TMP1
030612  104001                                  ERROR   1

030614  104415                                  SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.


030616  012704  000040                  1$:     MOV     #40,R4  ;THIS PATTERN WILL BE
030622  012702  177750                          MOV     #MAINT,R2       ;PUT IN THE MAINTENANCE
030626  005001                                  CLR     R1              ;REGISTER
030630  010015                                  MOV     R0,(R5)         ;PUT THE TEST PATTERN IN
                                                                        ;THE TEST ADDRESS
030632  000402                                  BR      64$

        030634                                  LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!
        030634                                  LOC=-4&LOC
        030640                                  LOC=LOC+4
        030640                                  .=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.
030640  010412                          64$:    MOV     R4,(R2)         ;TURN ON THE MAINT. REG.
030642  021500                                  CMP     (R5),R0         ;THE REFERENCE TO (R5)
```

```
030644  010112                                MOV     R1,(R2)          ;SHOULD CAUSE THE ERROR.

030646                               IIB3:
                                                                       ;THE ERROR DIDN'T OCCUR!
030646  010037  001636                        MOV     R0,$TMP2         ;REPORT FAILURE
030652  012737  031022  001640                MOV     #IIBT1,$TMP3
030660  005037  001642                        CLR     $TMP4
030664  104146                       64$:     ERROR   146

030666  012737  031064  000114       IIB4:    MOV     #IIBR2,@#CACHVEC          ;SET UP FOR THE GROUP ONE
030674  012737  030666  001510                MOV     #IIB4,$LPERR     ;ERROR
030702  012737  000044  177746                MOV     #S1M0,@#CONTRL   ;SELECT GROUP ONE

030710  012705  031022                        MOV     #IIBT1,R5        ;MAKE THE TEST ADDRESS A
030714  005715                                TST     (R5)             ;HIT, IN GROUP ONE.
030716  005715                                TST     (R5)

                                                                       ;SEE IF REFERENCE ADDRESS
030720  032737  000010  177752                BIT     #10,@#HITMIS     ;IS A HIT.
030726  001007                                BNE     1$
                                                                       ;IF NOT ERROR!
030730  010537  001636                        MOV     R5,$TMP2
030734  012737  000001  001634                MOV     #1,$TMP1
030742  104001                                ERROR   1

030744  104415                                SKIPT                    ;ERROR FATAL. GO TO NEXT TEST.


030746  012704  000200               1$:      MOV     #200,R4 ;THIS PATTERN WILL BE
030752  012702  177750                        MOV     #MAINT,R2        ;PUT IN THE MAINT. REG.
030756  005001                                CLR     R1
030760  010015                                MOV     R0,(R5)          ;PUT THE TEST PATTERN IN (R5),
                                                                       ;IIBT1.
030762  000402                                BR      50$              ;PUT THE NEXT INSTRUCTION EXECUTED
                                                                       ;ON AN EVEN WORD BOUNDARY SO THE
                                                                       ;SUBSEQUENT INSTRUCTION, A CMP,
                                                                       ;WILL BE A HIT.

        030764                                LOC=.            ;GET THE PC TO AN EVEN WORD BOUNDARY.!.
        030764                                LOC=-4&LOC
        030770                                LOC=LOC+4
        030770                                .=LOC

030770  000240                       50$:     NOP                      ;FOR SCOPING WITH AN OSCILLOSCOPE.
030772  010412                                MOV     R4,(R2)          ;TURN ON THE MAINT. REG.
030774  021500                                CMP     (R5),R0          ;THIS REFERENCE TO (R5) SHOULD
030776  010112                                MOV     R1,(R2)          ;CAUSE THE ERROR.

031000                               IIB5:
                                                                       ;THE ERROR DIDN'T OCCUR!
031000  010037  001636                        MOV     R0,$TMP2         ;REPORT FAILURE
031004  012737  031022  001640                MOV     #IIBT1,$TMP3
031012  005037  001642                        CLR     $TMP4
031016  104147                       64$:     ERROR   147

031020  000437                       IIB6:    BR      IIB7
```

```

031022  000000                          IIBT1:.WORD     0

031024                                  IIBR1:
031024  022737  004500  177744                  CMP     #4500,@#MEMERR    ;MAKE SURE THE ERROR
031032  001402                                  BEQ     2$                ;REGISTER IS SET PROPERLY
031034  000137  055440                  1$:     JMP     SPUR
031040  022737  031022  177740          2$:     CMP     #IIBT1,@#LOADRS   ;MAKE SURE THE ERROR
031046  001372                                  BNE     1$                ;OCCURRED AT THE CORRECT
                                                                          ;ADDRESS.
031050  022626                                  CMP     (SP)+,(SP)+       ;RESET THE STACK
031052  012737  177777  177744                  MOV     #-1,@#MEMERR      ;CLEAR THE ERROR REGISTERS.
031060  000137  030666                          JMP     IIB4              ;GO TEST GROUP ONE
031064                                  IIBR2:
031064  022737  004600  177744                  CMP     #4600,@#MEMERR    ;MAKE SURE THE ERROR
031072  001402                                  BEQ     2$                ;REGISTER IS SET PROPERLY
031074  000137  055440                  1$:     JMP     SPUR
031100  022737  031022  177740          2$:     CMP     #IIBT1,@#LOADRS   ;MAKE SURE THE ERROR
031106  001372                                  BNE     1$                ;OCCURRED AT THE CORRECT
                                                                          ;ADDRESS.
031110  022626                                  CMP     (SP)+,(SP)+       ;RESET THE STACK
031112  012737  177777  177744                  MOV     #-1,@#MEMERR      ;CLEAR THE ERROR REGISTERS.

031120  022700  177400                  IIB7:   CMP     #177400,R0                ;INCREMENT THE TEST
031124  001404                                  BEQ     IIB8              ;PATTERN
031126  062700  000400                          ADD     #400,R0
031132  000137  030520                          JMP     IIB1

031136  104414                          IIB8:   RSET


                                        ;;*****************************************************************
                                        ;*TEST 32        CACHE DATA MEMORY WORST CASE NOISE TEST
                                        ;*
                                        ;*THIS TEST DOES A GALLOPING 0'S AND 1'S OR PING PONG
                                        ;*TEST ON THE CACHE BIPOLAR DATA MEMORY.

                                        ;*
                                        ;;*****************************************************************
031140  000004                          TST32:  SCOPE
                                                                          ;SET THE SKAD REGISTER
031142  012737  032276  055572                  MOV     #TST33,SKAD       ;IN CASE THE TEST ABORTS.


031150  012737  055440  000114                  MOV     #SPUR,@#CACHVEC
031156  113737  001502  001632                  MOVB    $TSTNM,$TMP0      ;SAVE TESTN FOR PRINT OUT.

031164  005037  031666                          CLR     QQPAT1            ;BACK ROUND PATTERN OF
                                                                          ;0'S FOR THE GALLOPING
                                                                          ;1'S TEST TO BE EXECUTED
                                                                          ;FIRST.
031170  012737  000001  031662                  MOV     #1,QQFLG2         ;QQFLG=1 MEANS GALLOPING
                                                                          ;ONES TEST IN PROGRESS.
                                                                          ;QQFLG=0 MEANS GALLOPING
                                                                          ;ZEROES TEST IN PROGRESS.
031176  012737  031176  001510          QQ1:    MOV     #QQ1,$LPERR       ;SET ERROR LOOP INITIALLY
```

```
                                                                    ;TO THIS POINT.
031204  012737  000044  031676          MOV     #S1M0,QQGS          ;TEST GROUP 1 FIRST.
031212  012737  000030  031700          MOV     #S0M1,QQGM          ;S0M1 AND S1M0 ARE
                                                                    ;PATTERNS WHICH WILL BE
                                                                    ;LOADED INTO THE CACHE
                                                                    ;CONTROL REGISTER TO
                                                                    ;(SELECT GRP0 * MISS GRP1)
                                                                    ;AND (SELECT GRP1 * MISS GRP0)
                                                                    ;RESPECTIVELY.
031220  012737  000001  031664          MOV     #1,QQFLG1           ;QQFLG ONE CONTAINS THE
                                                                    ;NUMBER OF THE GROUP
                                                                    ;BEING TESTED, INITIALLY 1.

031226  012703  031226          QQ2:    MOV     #QQ2,R3             ;MAKE LOCATIONS QQ1
031232  012704  001000                  MOV     #1000,R4            ;THROUGH QQ2 + 2000 (OCT)
031236  013737  031700  177746  1$:     MOV     QQGM,@#CONTRL       ;HITS IN THE GROUP NOT
031244  005713                          TST     (R3)                ;BEING TESTED WHILE
031246  013737  031676  177746          MOV     QQGS,@#CONTRL       ;GETTING THESE LOCATIONS
031254  005763  002000                  TST     2000(R3)            ;TO BE MISSES IN THE
031260  062703  000002                  ADD     #2,R3               ;GROUP THAT IS BEING
031264  077414                          SOB     R4,1$               ;TESTED
031266  012704  001000                  MOV     #1000,R4            ;MAKE LOCATIONS TESTR2
031272  012705  142000                  MOV     #TESTR2,R5          ;THROUGH TESTR2+2000(OCT)
031276  013703  031676                  MOV     QQGS,R3             ;HITS IN THE GROUP
031302  042703  177717                  BIC     #177717,R3          ;BEING TESTED WHILE
031306  010337  177746                  MOV     R3,@#CONTRL         ;WRITING THE BACKGROUND
031312  013715  031666          QQ3:    MOV     QQPAT1,(R5)         ;PATTERN, IN QQPAT1, IN
031316  005715                          TST     (R5)
031320  005725                          TST     (R5)+               ;THEM. MAKE SURE THEY
031322  032737  000010  177752          BIT     #10,@#HITMIS        ;ARE HITS
031330  001011                          BNE     QQ4
031332  013737  031664  001634          MOV     QQFLG1,$TMP1        ;IF NOT ERROR
031340  010537  001636                  MOV     R5,$TMP2
031344  062737  177776  001636          ADD     #-2,$TMP2
031352  104001                  1$:     ERROR   1
031354  077422                  QQ4:    SOB     R4,QQ3
031356  013703  031700                  MOV     QQGM,R3             ;FROM NOW ON SELECT
031362  042703  177717                  BIC     #177717,R3          ;THE GROUP NOT BEING
031366  010337  177746                  MOV     R3,@#CONTRL         ;TESTED

031372  012704  031702                  MOV     #QQ10,R4            ;THE THREE ROUTINES
031376  042704  176000                  BIC     #176000,R4          ;QQ10-QQ11, QQ12-QQ13 AND
031402  012705  031756                  MOV     #QQ11,R5            ;QQ14-QQ15 ARE IDENTICAL
031406  042705  176000                  BIC     #176000,R5          ;EXCEPT FOR WHAT PART
031412  020405                          CMP     R4,R5               ;OF THE CACHE GROUP THAT
031414  002407                          BLT     QQ5                 ;IS NOT BEING TEST THEY
031416  012737  031760  031656          MOV     #QQ12,QQLO          ;LIE IN. HERE DECIDE
031424  012737  032036  031660          MOV     #QQ14,QQHI          ;WHICH TWO OF THE
031432  000450                          BR      QQ8                 ;ABOVE THREE IS APPROPRIATE
031434  012704  031760          QQ5:    MOV     #QQ12,R4            ;FOR THIS TEST.
031440  042704  176000                  BIC     #176000,R4
031444  012705  032034                  MOV     #QQ13,R5
031450  042705  176000                  BIC     #176000,R5
031454  020405                          CMP     R4,R5
031456  002407                          BLT     QQ6
031460  012737  032036  031656          MOV     #QQ14,QQLO
```

```
5781 031466 012737 031702 031660          MOV     #QQ10,QQHI
5782 031474 000427                        BR      QQ8
5783 031476 013704 031702         QQ6:    MOV     QQ10,R4
5784 031502 042704 176000                 BIC     #176000,R4
5785 031506 012705 031760                 MOV     #QQ12,R5
5786 031512 042705 176000                 BIC     #176000,R5
5787 031516 020405                        CMP     R4,R5
5788 031520 003007                        BGT     QQ7
5789 031522 012737 031702 031656          MOV     #QQ10,QQLO
5790 031530 012737 031760 031660          MOV     #QQ12,QQHI
5791 031536 000406                        BR      QQ8
5792 031540 012737 031760 031656  QQ7:    MOV     #QQ12,QQLO
5793 031546 012737 031702 031660          MOV     #QQ10,QQHI
5794
5795 031554 012702 142000         QQ8:    MOV     #TESTR2,R2        ;INITIALIZE FOR EITHER
5796 031560 012701 140000                 MOV     #TESTR1,R1        ;THE GALLOPING ONES OR
5797 031564 012705 001000                 MOV     #1000,R5          ;GALLOPING ZEROES TEST
5798                                                                ;WHICH IS PENDING.
5799 031570 012737 032200 000114          MOV     #QQERR1,@#CACHVEC         ;IF THE TEST FAILS A
5800                                                                ;PARITY ABORT IS LIKELY
5801                                                                ;SO SET UP TO GO THE
5802                                                                ;ERROR ROUTINE.
5803 031576 012737 031604 001510          MOV     #QQ9,$LPERR       ;SET THE LOOP ERROR
5804                                                                ;ADDRESS FOR THE BEGINNING
5805                                                                ;OF THE PASS ROUTINE.
5806
5807 031604 012703 142000         QQ9:    MOV     #TESTR2,R3        ;THIS DOES ONE PASS OF
5808 031610 012704 001000                 MOV     #1000,R4          ;THE TEST FOR EACH LOCATION.
5809 031614 005112                        COM     (R2)              ;PUT THE GALLOPING PATTERN
5810                                                                ;IN THE MEMORY.
5811
5812 031616 010100                QQ9.5:  MOV     R1,R0             ;SEE WHICH OF THE
5813 031620 042700 176000                 BIC     #176000,R0        ;TWO ROUTINES (QQ10,QQ12 OR
5814 031624 013737 031660 031670          MOV     QQHI,QQTMP1       ;QQ14) SHOULD FINISH
5815 031632 042737 176000 031670          BIC     #176000,QQTMP1    ;SETTING FOR TH'S TEST
5816 031640 020037 031670                 CMP     R0,QQTMP1         ;PASS.
5817 031644 002402                        BLT     1$
5818 031646 000177 000004                 JMP     @QQLO
5819 031652 000177 000002         1$:     JMP     @QQHI
5820
5821 031656 000000                QQLO:   .WORD   0                 ;QQLO AND QQHI CONTAIN THE
5822 031660 000000                QQHI:   .WORD   0                 ;ADDRESSES OF THE ROUTINES
5823                                                                ;TO BE USED IN SETTING UP
5824                                                                ;FOR A PASS.
5825 031662 000000                QQFLG2: .WORD   0                 ;1 IF DOING GALLOPING 1'S TEST.
5826                                                                ;0 IF DOING GALLOPING 0'S TEST.
5827 031664 000000                QQFLG1: .WORD   0                 ;GROUP BEING TESTED, 1 OR 0.
5828 031666 000000                QQPAT1: .WORD   0                 ;0 OR 1 BACKGROUND PATTERN.
5829 031670 000000                QQTMP1: .WORD   0                 ;USED AS TEMPORARY STORAGE.
5830 031672 000000                QQTMP2: .WORD   0
5831 031674 000000                QQTMP3: .WORD   0
5832 031676 000000                QQGS:   .WORD   0                 ;THESE REGISTERS HOLD PATTERNS
5833 031700 000000                QQGM:   .WORD   0                 ;WHICH ARE TO BE LOADED INTO THE
5834                                                                ;CACHE CONTROL REGISTER.
5835
5836                              ;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.
031702  000240                          QQ10:   NOP                         ;USED AS A BUFFER SO
031704  000240                                  NOP                         ;THIS CODE WON'T WIPE
                                                                            ;OUT DESIRED HITS
031706  012711  022312                          MOV     #022312,(R1)        ;020323=(CMP (R3)+,(R2)
031712  005711                                  TST     (R1)
031714  012761  077402  000002                  MOV     #077402,2(R1)       ;077402=(SOB R4,.-2)
031722  005761  000002                          TST     2(R1)
031726  012761  000137  000004                  MOV     #000137,4(R1)       ;000137=(JMP @#QQ16)
031734  005761  000004                          TST     4(R1)               ;QQ16
031740  012761  032114  000006                  MOV     #QQ16,6(R1)
031746  005761  000006                          TST     6(R1)
031752  000111                                  JMP     (R1)                ;GO DO A PASS.
031754  000240                                  NOP
031756  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.
031760  000240                          QQ12:   NOP                         ;USED AS A BUFFER SO
031762  000240                                  NOP                         ;THIS CODE WON'T WIPE
                                                                            ;OUT DESIRED HITS
031764  012711  022312                          MOV     #022312,(R1)        ;020323=(CMP (R3)+,(R2)
031770  005711                                  TST     (R1)
031772  012761  077402  000002                  MOV     #077402,2(R1)       ;077402=(SOB R4,.-2)
032000  005761  000002                          TST     2(R1)
032004  012761  000137  000004                  MOV     #000137,4(R1)       ;000137=(JMP @#QQ16)
032012  005761  000004                          TST     4(R1)               ;QQ16
032016  012761  032114  000006                  MOV     #QQ16,6(R1)
032024  005761  000006                          TST     6(R1)
032030  000111                                  JMP     (R1)                ;GO DO A PASS.
032032  000240                                  NOP
032034  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.
032036  000240                          QQ14:   NOP                         ;USED AS A BUFFER SO
032040  000240                                  NOP                         ;THIS CODE WON'T WIPE
                                                                            ;OUT DESIRED HITS
032042  012711  022312                          MOV     #022312,(R1)        ;020323=(CMP (R3)+,(R2)
032046  005711                                  TST     (R1)
032050  012761  077402  000002                  MOV     #077402,2(R1)       ;077402=(SOB R4,.-2)
032056  005761  000002                          TST     2(R1)
032062  012761  000137  000004                  MOV     #000137,4(R1)       ;000137=(JMP @#QQ16)
032070  005761  000004                          TST     4(R1)               ;QQ16
032074  012761  032114  000006                  MOV     #QQ16,6(R1)
032102  005761  000006                          TST     6(R1)
032106  000111                                  JMP     (R1)                ;GO DO A PASS.
```

```
032110  000240                          NOP
032112  000240                  QQ15:   NOP

032114  005122                  QQ16:   COM     (R2)+                   ;PASS DONE. RESTORE THE
                                                                        ;BACKGROUND PATTERN.

032116  062701  000002          QQ17:   ADD     #2,R1                   ;GO TO NEXT LOCATION FOR
                                                                        ;NEXT PASS.
032122  005305                          DEC     R5                      ;DO ANOTHER PASS?
032124  001402                          BEQ     1$
032126  000137  031604                  JMP     QQ9
032132                          1$:
032132  012737  000044  031700          MOV     #S1M0,QQGM              ;TESTED GROUP 1 NOW GO BACK
032140  012737  000030  031676          MOV     #S0M1,QQGS              ;AND TEST GROUP 0
032146  005337  031664                  DEC     QQFLG1
032152  001002                          BNE     QQ18
032154  000137  031226                  JMP     QQ2

032160  012737  177777  031666  QQ18:   MOV     #-1,QQPAT1              ;GALLOPING 1'S TEST IS
032166  005337  031662                  DEC     QQFLG2                  ;COMPLETE, ON BOTH GROUPS,
032172  001041                          BNE     QQ19                    ;SET THE BACKGROUND PATTERN
032174  000137  031176                  JMP     QQ1                     ;FOR GALLOPING 0'S AND GO
                                                                        ;BACK TO PERFORM THIS TEST
                                                                        ;ON BOTH GROUPS.

032200  013737  177744  001634  QQERR1: MOV     @#MEMERR,$TMP1          ;COME HERE IF DURING THE
032206  013737  177740  001636          MOV     @#LOADRS,$TMP2          ;TEST A TRAP OR ABORT
032214  013737  177742  001640          MOV     @#HIADRS,$TMP3          ;OCCURRED TO CACHVEC
032222  011637  001642                  MOV     (SP),$TMP4
032226  022626                          CMP     (SP)+,(SP)+
032230  010137  001644                  MOV     R1,$TMP5
032234  013737  031664  001646          MOV     QQFLG1,$TMP6
032242  032737  000600  001634          BIT     #600,$TMP1
032250  001002                          BNE     QQERR2
032252  104002                          ERROR   2
032254  000406                          BR      QQERR4
032256  005737  031666          QQERR2: TST     QQPAT1                  ;GALLOPING 1' OR 0'S?
032262  001002                          BNE     QQERR3
032264  104003                          ERROR   3                       ;0'S.
032266  000401                          BR      QQERR4
032270  104004                  QQERR3: ERROR   4                       ;1'S
032272  000137  032114          QQERR4: JMP     QQ16                    ;CONTINUE?

032276                          QQ19:                                   ;DONE! PERHAPS PRINT SUMMARY.
                                                                        ;?????

                                ;;*****************************************************************
                                ;*TEST 33        CACHE DATA MEMORY CHIP SELECTION LOGIC TEST
                                ;*
                                ;*THIS ROUTINE TESTS THE 'CHIP-SET' ENABLE LOGIC FOR THE CACHE DATA
                                ;*MEMORY.  TO DEFINE THE TERM 'CHIP-SET' CONSIDER THE CACHE MEMORY AS
                                ;*BEING DIVIDED INTO FOUR SETS OF 256 (DEC) X 1 BIT BIPOLAR MEMORY
                                ;*CHIPS.  EACH SET IS MADE UP OF 18 CHIPS, THE 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).
                                ;*
                                ;;*************************************************************************
032276  000004                  TST33:  SCOPE
032300  012737  000040  001702          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                                                ;SET THE SKAD REGISTER
032306  012737  034042  055572          MOV     #TST34,SKAD     ;IN CASE THE TEST ABORTS.


032314  113737  001502  001632          MOVB    $TSTNM,$TMPO    ;PUT THE TEST NUMBER IN
                                                                ;$TMPO FOR PRINT OUT.
032322  012737  055440  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.

032330  012737  000014  177746  KK1:    MOV     #MOM1,@#CONTRL  ;FORCE MISSES AND
032336  005037  033676                  CLR     KKPAT1          ;INITIALIZE THE TEST PATTERN
032342  012737  177777  033700          MOV     #177777,KKPAT2  ;TABLE
032350  012737  125252  033/02          MOV     #125252,KKPAT3
032356  012737  052525  033704          MOV     #52525,KKPAT4

032364  005037  033672                  CLR     KKFLG1          ;INITIALIZE KKFLG1:
                                                                ;0 MEANS WRITE PATTERNS IN
                                                                ;IN THE UPWARD DIRECTION
                                                                ;1 MEANS WRITE PATTERNS IN
                                                                ;THE DOWNWARD DIRECTION

032370  012700  033712          KK2:    MOV     #KKTMP2,R0      ;ESTABLISH AN OFFSET FOR
032374  042700  176003                  BIC     #176003,R0      ;A TEST AREA WHOSE HITS
                                                                ;WILL NOT BE INTERFERRED WITH BY
032400  010001                          MOV     R0,R1           ;THE CYCLES CAUSED WHILE
032402  062701  140000                  ADD     #TESTR1,R1      ;FETCHING THE TEST CODE.
032406  010002                          MOV     R0,R2
032410  062702  142000                  ADD     #TESTR2,R2

032414  010137  001644                  MOV     R1,$TMP5        ;SAVE THE ADDRESSES OF
032420  010137  001646                  MOV     R1,$TMP6        ;THE FOUR TEST WORD LOCATIONS,
032424  062737  000002  001646          ADD     #2,$TMP6        ;FOR TYPE OUT IN CASE
032432  010237  001650                  MOV     R2,$TMP7        ;OF ERROR.
032436  010237  001652                  MOV     R2,$TMP10
032442  062737  000002  001652          ADD     #2,$TMP10

032450  012705  033700                  MOV     #KKPAT2,R5      ;A POINTER USED IN GENERATING
                                                                ;EVERY PERMUTATION OF THE TEST
                                                                ;PATTERNS.
032454  012700  000006                  MOV     #6,R0           ;R0 AND KKCNT1 ARE ALSO USED
032460  012737  000004  033674          MOV     #4,KKCNT1       ;IN GENERATING THE PERMUTATIONS.

032466  012737  032474  001510          MOV     #KK3,$LPERR     ;WHEN LOOPING ON ERROR GO TO KK3.
```

```
6005  032474  000240                         KK3:    NOP     ;FOR SCOPING PER POSES
6006  032476  012737  000034  177746                 MOV     #S0M0M1,@#CONTRL;MAKE THE TEST AREA HITS
6007  032504  005711                                 TST     (R1)            ;IN THE CACHE GROUPS.
6008  032506  005761  000002                         TST     2(R1)
6009  032512  012737  000054  177746                 MOV     #S1M0M1,@#CONTRL
6010  032520  005712                                 TST     (R2)
6011  032522  005762  000002                         TST     2(R2)
6012  032526  005037  177746                         CLR     @#CONTRL
6013
6014
6015  032532  005711                                 TST     (R1)
6016
6017                                                                         ;SEE IF REFERENCE ADDRESS
6018  032534  032737  000010  177752                 BIT     #10,@#HITMIS    ;IS A HIT.
6019  032542  001006                                 BNE     1$
6020                                                                         ;IF NOT ERROR.
6021  032544  010137  001636                         MOV     R1,$TMP2
6022  032550  012737  000000  001634                 MOV     #0,$TMP1
6023  032556  104001                                 ERROR   1
6024
6025
6026
6027  032560                                 1$:
6028
6029  032560  005761  000002                         TST     2(R1)
6030
6031                                                                         ;SEE IF REFERENCE ADDRESS
6032  032564  032737  000010  177752                 BIT     #10,@#HITMIS    ;IS A HIT.
6033  032572  001011                                 BNE     2$
6034                                                                         ;IF NOT ERROR!
6035  032574  010137  001636                         MOV     R1,$TMP2
6036  032600  062737  000002  001636                 ADD     #2,$TMP2
6037  032606  012737  000000  001634                 MOV     #0,$TMP1
6038  032614  104001                                 ERROR   1
6039
6040
6041
6042  032616                                 2$:
6043
6044  032616  005712                                 TST     (R2)
6045
6046                                                                         ;SEE IF REFERENCE ADDRESS
6047  032620  032737  000010  177752                 BIT     #10,@#HITMIS    ;IS A HIT.
6048  032626  001006                                 BNE     3$
6049                                                                         ;IF NOT ERROR!
6050  032630  010237  001636                         MOV     R2,$TMP2
6051  032634  012737  000001  001634                 MOV     #1,$TMP1
6052  032642  104001                                 ERROR   1
6053
6054
6055
6056  032644                                 3$:
6057
6058  032644  005762  000002                         TST     2(R2)
6059
6060                                                                         ;SEE IF REFERENCE ADDRESS
```

```
032650  032737  000010  177752            BIT     #10,@#HITMIS    ;IS A HIT.
032656  001011                            BNE     4$
                                                                  ;IF NOT ERROR!
032660  010237  001636                    MOV     R2,$TMP2
032664  062737  000002  001636            ADD     #2,$TMP2
032672  012737  000001  001634            MOV     #1,$TMP1
032700  104001                            ERROR   1




032702  005737  033672          4$:       TST     KKFLG1          ;SEE IF THE TST PATTERN
                                                                  ;SHOULD BE WRITTEN UPWARD
                                                                  ;OR DOWNWARD.
032706  001045                            BNE     KK4             ;BRANCH IF DOWNWARD
                                                                  ;OTHERWISE WRITE IT IN THE
                                                                  ;UPWARD DIRECTION.
032710  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
032716  013703  033676                    MOV     KKPAT1,R3               ;LOCATION KKPAT1, INTO THE
032722  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 0
032726  010361  000000                    MOV     R3,0(R1)
032732  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
032740  013703  033700                    MOV     KKPAT2,R3               ;LOCATION KKPAT2, INTO THE
032744  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 2
032750  010361  000002                    MOV     R3,2(R1)
032754  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
032762  013703  033702                    MOV     KKPAT3,R3               ;LOCATION KKPAT3, INTO THE
032766  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 0
032772  010362  000000                    MOV     R3,0(R2)
032776  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
033004  013703  033704                    MOV     KKPAT4,R3               ;LOCATION KKPAT4, INTO THE
033010  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 2
033014  010362  000002                    MOV     R3,2(R2)
033020  000444                            BR      KK5
033022                          KK4:                              ;WRITE THE PATTERN IN THE
                                                                  ;DOWNWARD DIRECTION
033022  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
033030  013703  033704                    MOV     KKPAT4,R3               ;LOCATION KKPAT4, INTO THE
033034  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 2
033040  010362  000002                    MOV     R3,2(R2)
033044  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
033052  013703  033702                    MOV     KKPAT3,R3               ;LOCATION KKPAT3, INTO THE
033056  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R2 PLUS 0
033062  010362  000000                    MOV     R3,0(R2)
033066  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
033074  013703  033700                    MOV     KKPAT2,R3               ;LOCATION KKPAT2, INTO THE
033100  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 2
033104  010361  000002                    MOV     R3,2(R1)
033110  012737  000014  177746            MOV     #MOM1,@#CONTRL  ;WRITE THE TEST PATTERN, FROM
033116  013703  033676                    MOV     KKPAT1,R3               ;LOCATION KKPAT1, INTO THE
033122  005037  177746                    CLR     @#CONTRL        ;ADDRESS IN R1 PLUS 0
033126  010361  000000                    MOV     R3,0(R1)

033132                          KK5:
033132  012737  000014  177746            MOV     #MOM1,@#CONTRL
033140  013703  033676                    MOV     KKPAT1,R3               ;SEE IF THE TEST PATTERN WAS
```

```
033144  005037  177746                  CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
033150  016104  000000                  MOV     0(R1),R4


                                                                ;SEE IF REFERENCE ADDRESS
033154  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
033162  001006                          BNE     64$
                                                                ;IF NOT ERROR!
033164  010137  001636                  MOV     R1,$TMP2
033170  012737  000000  001634          MOV     #0,$TMP1
033176  104001                          ERROR   1


033200  020403                  64$:    CMP     R4,R3
033202  001402                          BEQ     KK6
033204  004737  033722                  JSR     PC,KKERR1

033210                          KK6:
033210  012737  000014  177746          MOV     #MOM1,@#CONTRL
033216  013703  033700                  MOV     KKPAT2,R3              ;SEE IF THE TEST PATTERN WAS
033222  005037  177746                  CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
033226  016104  000002                  MOV     2(R1),R4


                                                                ;SEE IF REFERENCE ADDRESS
033232  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
033240  001011                          BNE     64$
                                                                ;IF NOT ERROR!
033242  010137  001636                  MOV     R1,$TMP2
033246  062737  000002  001636          ADD     #2,$TMP2
033254  012737  000000  001634          MOV     #0,$TMP1
033262  104001                          ERROR   1


033264  020403                  64$:    CMP     R4,R3
033266  001402                          BEQ     KK7
033270  004737  033734                  JSR     PC,KKERR2

033274                          KK7:
033274  012737  000014  177746          MOV     #MOM1,@#CONTRL
033302  013703  033702                  MOV     KKPAT3,R3              ;SEE IF THE TEST PATTERN WAS
033306  005037  177746                  CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
033312  016204  000000                  MOV     0(R2),R4


                                                                ;SEE IF REFERENCE ADDRESS
033316  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
033324  001006                          BNE     64$
                                                                ;IF NOT ERROR!
033326  010237  001636                  MOV     R2,$TMP2
033332  012737  000001  001634          MOV     #1,$TMP1
033340  104001                          ERROR   1


033342  020403                  64$:    CMP     R4,R3
033344  001402                          BEQ     KK8
033346  004737  033754                  JSR     PC,KKERR3

033352                          KK8:
```

```
033352  012737  000014  177746           MOV     #MOM1,@#CONTRL
033360  013703  033704                   MOV     KKPAT4,R3               ;SEE IF THE TEST PATTERN WAS
033364  005037  177746                   CLR     @#CONTRL        ;WRITTEN OR IS READ CORRECTLY.
033370  016204  000002                   MOV     2(R2),R4

                                                                 ;SEE IF REFERENCE ADDRESS
033374  032737  000010  177752           BIT     #10,@#HITMIS    ;IS A HIT.
033402  001011                           BNE     64$
                                                                 ;IF NOT ERROR!
033404  010237  001636                   MOV     R2,$TMP2
033410  062737  000002  001636           ADD     #2,$TMP2
033416  012737  000001  001634           MOV     #1,$TMP1
033424  104001                           ERROR   1


033426  020403                  64$:     CMP     R4,R3
033430  001402                           BEQ     KK10
033432  004737  033770                   JSR     PC,KKERR4

033436  005737  033672          KK10:    TST     KKFLG1          ;SEE IF THIS PERMUTATION OF
033442  001005                           BNE     KK11            ;THE TEST PATTERN HAS BEEN
033444  012737  177777  033672           MOV     #-1,KKFLG1      ;WRITTEN BOTH UPWARD AND
033452  000137  032474                   JMP     KK3             ;DOWNWARD. IF NOT, KKFLG IS 0.
                                                                 ;GO BACK TO WRITE IT DOWNWARD.

033456  005037  033672          KK11:    CLR     KKFLG1          ;GENERATE THE NEXT PERMUTATION
033462  012737  000014  177746           MOV     #MOM1,@#CONTRL  ;OF THE TEST PATTERN IN THE
                                                                 ;TEST TABLE
033470  020527  033704                   CMP     R5,#KKPAT4
033474  001011                           BNE     KK12

033476  011537  033706                   MOV     (R5),KKPAT5
033502  013715  033700                   MOV     KKPAT2,(R5)
033506  012705  033700                   MOV     #KKPAT2,R5
033512  013715  033706                   MOV     KKPAT5,(R5)
033516  000406                           BR      KK13

033520  012537  033706          KK12:    MOV     (R5)+,KKPAT5
033524  011565  177776                   MOV     (R5),-2(R5)
033530  013715  033706                   MOV     KKPAT5,(R5)

033534  005300                  KK13:    DEC     R0
033536  001402                           BEQ     KK14
033540  000137  032474                   JMP     KK3             ;GO DO NEXT PERMUTATION.

033544  012700  000006          KK14:    MOV     #6,R0
033550  013737  033676  033706           MOV     KKPAT1,KKPAT5
033556  005337  033674                   DEC     KKCNT1

033562  022737  000003  033674           CMP     #3,KKCNT1
033570  001010                           BNE     KK15

033572  013737  033700  033676           MOV     KKPAT2,KKPAT1
033600  013737  033706  033700           MOV     KKPAT5,KKPAT2
033606  000137  032474                   JMP     KK3             ;GO DO NEXT PERMUTATION.

```

```
033612  022737  000002  033674  KK15:     CMP     #2,KKCNT1
033620  001010                            BNE     KK16

033622  013737  033702  033676            MOV     KKPAT3,KKPAT1
033630  013737  033706  033702            MOV     KKPAT5,KKPAT3
033636  000137  032474                    JMP     KK3             ;GO DO NEXT PERMUTION.

033642  022737  000001  033674  KK16:     CMP     #1,KKCNT1
033650  001073                            BNE     KK17            ;BRANCH IF DONE.

033652  013737  033704  033676            MOV     KKPAT4,KKPAT1
033660  013737  033706  033704            MOV     KKPAT5,KKPAT4
033666  000137  032474                    JMP     KK3             ;GO DO NEXT PERMUTATION.


033672  000000                  KKFLG1:   .WORD   0               ;0 IF STORING PATTERN UPWARD
                                                                  ;1 IF STORING DOWNWARD.

033674  000000                  KKCNT1:   .WORD   0               ;COUNTER USED IN GENERATING
                                                                  ;THE TEST PATTERN PERMUTATIONS.

033676  000000                  KKPAT1:   .WORD   0               ;TEST PATTERN TABLE.
033700  000000                  KKPAT2:   .WORD   0
033702  000000                  KKPAT3:   .WORD   0
033704  000000                  KKPAT4:   .WORD   0
033706  000000                  KKPAT5:   .WORD   0

033710  000000                  KKTMP1:   .WORD   0               ;USED TO LOCATE A TEST AREA WHOSE
033712  000000  000000  000000  KKTMP2:   .WORD   0,0,0,0         ;HITS WON'T BE WIPED OUT BY TEST CODE.
033720  000000

033722  010137  001642          KKERR1:   MOV     R1,$TMP4        ;ERROR REPORTING ROUTINES
033726  005037  001640                    CLR     $TMP3
033732  000427                            BR      KKERR5

033734  010137  001642          KKERR2:   MOV     R1,$TMP4
033740  062737  000002  001642            ADD     #2,$TMP4
033746  005037  001640                    CLR     $TMP3
033752  000417                            BR      KKERR5

033754  010237  001642          KKERR3:   MOV     R2,$TMP4
033760  013737  000001  001640            MOV     1,$TMP3
033766  000411                            BR      KKERR5

033770  010237  001642          KKERR4:   MOV     R2,$TMP4
033774  062737  000002  001642            ADD     #2,$TMP4
034002  012737  000001  001640            MOV     #1,$TMP3
034010  000400                            BR      KKERR5

034012  010337  001636          KKERR5:   MOV     R3,$TMP2
034016  011637  001634                    MOV     (SP),$TMP1
034022  012737  000014  177746            MOV     #MOM1,@#CONTRL

034030  104021                            ERROR   21

034032  005037  177746                    CLR     @#CONTRL
```

```
034036  000207                          RTS     PC

034040  104414                  KK17:   RSET                      ;DONE.

                                ;;*************************************************************
                                ;*TEST 34       CACHE DATA MEMORY BYTE ENABLE LOGIC TEST
                                ;*
                                ;*THIS TEST PERFORMS A CHECK OF THE BYTE ENABLE LOGIC
                                ;*IN THE CACHE DATA MEMORY. THE BYTE PATTERNS 1, 2, 4, 10, 20,
                                ;*40, 100 A 200 ARE USED. THE FIRST FOUR PATTERNS ARE WRITTEN
                                ;*IN CONSECUTIVE BYTE LOCATIONS WHICH ARE HITS IN GROUP 0.
                                ;*THE REMAINING FOUR PATTERNS ARE WRITTEN IN CONSECUTIVE
                                ;*BYTE LOCATIONS WHICH ARE HITS IN GROUP 1. EACH PATTERN IS
                                ;*READ BACK CHECKED AND THE COMPLIMENT PATTERN IS WRITTEN.
                                ;*AFTER ALL THE PATTERNS HAVE BEEN CHECKED AND COMPLEMENTED
                                ;*THE COMPLIMENTED PATTERNS ARE CHECKED.
                                ;*
                                ;;*************************************************************
034042  000004                  TST34:  SCOPE
034044  012737  000040  001702          MOV     #40,$TIMES        ;;DO 40 ITERATIONS
                                                                  ;SET THE SKAD REGISTER
034052  012737  035704  055572          MOV     #TST35,SKAD       ;IN CASE THE TEST ABORTS.

034060  012737  055440  000114          MOV     #SPUR,@#CACHVEC   ;ADDRESS AND PUT THE NO ERROR
034066  113737  001502  001632          MOVB    $TSTNM,$TMP0      ;EXPECTED ROUTINES ADDRESS IN
                                                                  ;THE PARITY ERROR VECTOR.

034074  012737  001001  035542  MM1:    MOV     #001001,MMPAT1    ;SET UP THE PATTERN
034102  012737  004004  035544          MOV     #004004,MMPAT2    ;REGISTERS.
034110  012737  020020  035546          MOV     #020020,MMPAT3
034116  012737  100100  035550          MOV     #100100,MMPAT4

034124  012700  035554                  MOV     #MMTMP2,R0        ;LOCATE THE TEST AREA IN
034130  042700  176003                  BIC     #176003,R0        ;MEMORY WHOSE 'HITS' WILL NOT
034134  010001                          MOV     R0,R1             ;INTERFER WITH HITS CAUSED
034136  062701  140000                  ADD     #TESTR1,R1        ;BY EXECUTING THIS TEST'S
034142  010002                          MOV     R0,R2             ;CODE.
034144  062702  142000                  ADD     #TESTR2,R2

034150  010137  001644                  MOV     R1,$TMP5          ;SAVE THE TEST AREA ADDRESSES
034154  010137  001646                  MOV     R1,$TMP6                  ;FOR ERROR PRINT OUT.
034160  062737  000002  001646          ADD     #2,$TMP6
034166  010237  001650                  MOV     R2,$TMP7
034172  010237  001652                  MOV     R2,$TMP10
034176  062737  000002  001652          ADD     #2,$TMP10

034204  012737  034212  001510          MOV     #MM2,$LPERR       ;SET THE LOOP ON ERROR REGISTER.

034212  000240                  MM2:    NOP
034214  012737  000034  177746          MOV     #S0M0M1,@#CONTRL          ;MAKE THE TEST AREAS HITS
034222  005711                          TST     (R1)              ;IN GROUP 0 AND 1.
034224  005761  000002                  TST     2(R1)
034230  012737  000054  177746          MOV     #S1M0M1,@#CONTRL
034236  005712                          TST     (R2)
034240  005762  000002                  TST     2(R2)
034244  005037  177746                  CLR     @#CONTRL
```

```


034250  005711                          TST     (R1)

                                                                ;SEE IF REFERENCE ADDRESS
034252  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
034260  001006                          BNE     MM3
                                                                ;IF NOT ERROR!
034262  010137  001636                  MOV     R1,$TMP2
034266  012737  000000  001634          MOV     #0,$TMP1
034274  104001                          ERROR   1



034276                          MM3:

034276  005761  000002                  TST     2(R1)

                                                                ;SEE IF REFERENCE ADDRESS
034302  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
034310  001011                          BNE     MM4
                                                                ;IF NOT ERROR!
034312  010137  001636                  MOV     R1,$TMP2
034316  062737  000002  001636          ADD     #2,$TMP2
034324  012737  000000  001634          MOV     #0,$TMP1
034332  104001                          ERROR   1



034334                          MM4:

034334  005712                          TST     (R2)

                                                                ;SEE IF REFERENCE ADDRESS
034336  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
034344  001006                          BNE     MM5
                                                                ;IF NOT ERROR!
034346  010237  001636                  MOV     R2,$TMP2
034352  012737  000001  001634          MOV     #1,$TMP1
034360  104001                          ERROR   1



034362                          MM5:

034362  005762  000002                  TST     2(R2)

                                                                ;SEE IF REFERENCE ADDRESS
034366  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
034374  001014                          BNE     MM6
                                                                ;IF NOT ERROR!
034376  010237  001636                  MOV     R2,$TMP2
034402  062737  000002  001636          ADD     #2,$TMP2
034410  012737  000001  001634          MOV     #1,$TMP1
034416  104001                          ERROR   1

```

```


034420  012737  034426  001510          MOV     #MM6,$LPERR     ;SET LOOP ON ERROR ADDRESS
034426  012703  000001          MM6:    MOV     #1,R3
034432  012704  000004                  MOV     #4,R4
034436  110321                  MM7:    MOVB    R3,(R1)+        ;PUT THE TEST PATTERN
034440  006103                          ROL     R3              ;IN GROUP 0
034442  077403                          SOB     R4,MM7

034444  012704  000004                  MOV     #4,R4
034450  110322                  MM8:    MOVB    R3,(R2)+        ;PUT THE TEST PATTERN
034452  006103                          ROL     R3              ;IN GROUP 1
034454  077403                          SOB     R4,MM8
034456  010001                          MOV     R0,R1
034460  062701  140000                  ADD     #TESTR1,R1      ;RE-ESTABLISH POINTERS TO
034464  010002                          MOV     R0,R2           ;THE TEST LOCATIONS.
034466  062702  142000                  ADD     #TESTR2,R2
034472  012703  035542                  MOV     #MMPAT1,R3      ;PUT THE ADDRESS OF THE TEST
                                                                ;PATTERN REGISTERS IN R3

034476  005005                          CLR     R5


034500  005005                          CLR     R5
034502  111105                          MOVB    (R1),R5         ;GET THE PATTERN OUT OF
034504  032737  000010  177752          BIT     #10,@#HITMIS    ;THIS BYTE MAKING SURE IT
034512  001006                          BNE     MM9             ;IS A HIT
034514  010137  001636                  MOV     R1,$TMP2
034520  012737  000000  001634          MOV     #0,$TMP1
034526  104001                          ERROR   1

034530  042705  177400          MM9:    BIC     #177400,R5
034534  022705  000001                  CMP     #1,R5   ;SEE IF THE DATA IS CORRECT.
034540  001402                          BEQ     MM10
034542  004737  035564                  JSR     PC,MMERR1
034546  105121                  MM10:   COMB    (R1)+           ;COMPLIMENT THE TEST PATTERN
034550  012713  001376                  MOV     #001376,(R3)



034554  005005                          CLR     R5
034556  111105                          MOVB    (R1),R5         ;GET THE PATTERN OUT OF
034560  032737  000010  177752          BIT     #10,@#HITMIS    ;THIS BYTE MAKING SURE IT
034566  001006                          BNE     MM11            ;IS A HIT
034570  010137  001636                  MOV     R1,$TMP2
034574  012737  000000  001634          MOV     #0,$TMP1
034602  104001                          ERROR   1

034604  042705  177400          MM11:   BIC     #177400,R5
034610  022705  000002                  CMP     #2,R5   ;SEE IF THE DATA IS CORRECT.
034614  001402                          BEQ     MM12
034616  004737  035564                  JSR     PC,MMERR1
034622  105121                  MM12:   COMB    (R1)+           ;COMPLIMENT THE TEST PATTERN
034624  012713  176776                  MOV     #176776,(R3)


```

```
034630  062703  000002                    ADD     #2,R3               ;POINT TO THE NEXT ELEMENT
                                                                      ;IN THE TEST PATTERN TABLE.

034634  005005                            CLR     R5
034636  111105                            MOVB    (R1),R5             ;GET THE PATTERN OUT OF
034640  032737  000010  177752            BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034646  001006                            BNE     MM13                ;IS A HIT
034650  010137  001636                    MOV     R1,$TMP2
034654  012737  000000  001634            MOV     #0,$TMP1
034662  104001                            ERROP   1

034664  042705  177400          MM13:     BIC     #177400,R5
034670  022705  000004                    CMP     #4,R5   ;SEE IF THE DATA IS CORRECT.
034674  001402                            BEQ     MM14
034676  004737  035564                    JSR     PC,MMERR1
034702  105121                  MM14:     COMB    (R1)+               ;COMPLIMENT THE TES~ PATTERN
034704  012713  004373                    MOV     #004373,(R3)



034710  005005                            CLR     R5
034712  111105                            MOVB    (R1),R5             ;GET THE PATTERN OUT OF
034714  032737  000010  177752            BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
034722  001006                            BNE     MM15                ;IS A HIT
034724  010137  001636                    MOV     R1,$TMP2
034730  012737  000000  001634            MOV     #0,$TMP1
034736  104001                            ERROR   1

034740  042705  177400          MM15:     BIC     #177400,R5
034744  022705  000010                    CMP     #10,R5  ;SEE IF THE DATA IS CORRECT.
034750  001402                            BEQ     MM16
034752  004737  035564                    JSR     PC,MMERR1
034756  105121                  MM16:     COMB    (R1)+               ;COMPLIMENT THE TEST PATTERN
034760  012713  173773                    MOV     #173773,(R3)


034764  062703  000002                    ADD     #2,R3               ;POINT TO THE NEXT ELEMENT
                                                                      ;IN THE TEST PATTERN TABLE.

034770  005005                            CLR     R5
034772  111205                            MOVB    (R2),R5             ;GET THE PATTERN OUT OF
034774  032737  000010  177752            BIT     #10,@#HITMIS        ;THIS BYTE MAKING SURE IT
035002  001006                            BNE     MM17                ;IS A HIT
035004  010237  001636                    MOV     R2,$TMP2
035010  012737  000001  001634            MOV     #1,$TMP1
035016  104001                            ERROR   1

035020  042705  177400          MM17:     BIC     #177400,R5
035024  022705  000020                    CMP     #20,R5  ;SEE IF THE DATA IS CORRE^T.
035030  001402                            BEQ     MM18
035032  004737  035576                    JSR     PC,MMERR2
035036  105122                  MM18:     COMB    (R2)+               ;COMPLIMENT THE TEST PATTERN
035040  012713  020357                    MOV     #020357,(R3)



```

```
035044  005005                                  CLR     R5
035046  111205                                  MOVB    (R2),R5          ;GET THE PATTERN OUT OF
035050  032737  000010  177752                  BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
035056  001006                                  BNE     MM19             ;IS A HIT
035060  010237  001636                          MOV     R2,$TMP2
035064  012737  000001  001634                  MOV     #1,$TMP1
035072  104001                                  ERROR   1

035074  042705  177400                  MM19:   BIC     #177400,R5
035100  022705  000040                          CMP     #40,R5  ;SEE IF THE DATA IS CORRECT.
035104  001402                                  BEQ     MM20
035106  004737  035576                          JSR     PC,MMERR2
035112  105122                          MM20:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
035114  012713  157757                          MOV     #157757,(R3)


035120  062703  000002                          ADD     #2,R3            ;POINT TO THE LAST ELEMENT
                                                                         ;IN THE TEST PATTERN TABLE.

035124  005005                                  CLR     R5
035126  111205                                  MOVB    (R2),R5          ;GET THE PATTERN OUT OF
035130  032737  000010  177752                  BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
035136  001006                                  BNE     MM21             ;IS A HIT
035140  010237  001636                          MOV     R2,$TMP2
035144  012737  000001  001634                  MOV     #1,$TMP1
035152  104001                                  ERROR   1

035154  042705  177400                  MM21:   BIC     #177400,R5
035160  022705  000100                          CMP     #100,R5 ;SEE IF THE DATA IS CORRECT.
035164  001402                                  BEQ     MM22
035166  004737  035576                          JSR     PC,MMERR2
035172  105122                          MM22:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
035174  012713  100277                          MOV     #100277,(R3)



035200  005005                                  CLR     R5
035202  111205                                  MOVB    (R2),R5          ;GET THE PATTERN OUT OF
035204  032737  000010  177752                  BIT     #10,@#HITMIS     ;THIS BYTE MAKING SURE IT
035212  001006                                  BNE     MM23             ;IS A HIT
035214  010237  001636                          MOV     R2,$TMP2
035220  012737  000001  001634                  MOV     #1,$TMP1
035226  104001                                  ERROR   1

035230  042705  177400                  MM23:   BIC     #177400,R5
035234  022705  000200                          CMP     #200,R5 ;SEE IF THE DATA IS CORRECT.
035240  001402                                  BEQ     MM24
035242  004737  035576                          JSR     PC,MMERR2
035246  105122                          MM24:   COMB    (R2)+            ;COMPLIMENT THE TEST PATTERN
035250  012713  077677                          MOV     #077677,(R3)


035254  010001                                  MOV     R0,R1            ;RE-ESTABLISH POINTERS TO
035256  062701  140000                          ADD     #TESTR1,R1       ;THE TEST AREA
035262  010002                                  MOV     R0,R2
035264  062702  142000                          ADD     #TESTR2,R2
```

```


035270  012105                                  MOV     (R1)+,R5        ;CHECK THE COMPLIMENTED

035272  005761  177776                          TST     -2(R1)

                                                                        ;SEE IF REFERENCE ADDRFSS
035276  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
035304  0C1011                                  BNE     MM25
                                                                        ;IF NOT ERROR.
035306  010137  001636                          MOV     R1,$TMP2
035312  062737  177776  001636                  ADD     #-2,$TMP2
035320  012737  000000  001634                  MOV     #0,$TMP1
035326  104001                                  ERROR   1




035330  020537  035542          MM25:           CMP     R5,MMPAT1               ;IS PATTERN CORRECT?
035334  001402                                  BEQ     MM26
035336  004737  035626                          JSR     PC,MMERR4


035342                          MM26:

035342  012105                                  MOV     (R1)+,R5        ;CHECK THE COMPLIMENTFD

035344  005761  177776                          TST     -2(R1)

                                                                        ;SEE IF REFERENCE ADDRESS
035350  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
035356  001011                                  BNE     MM27
                                                                        ;IF NOT ERROR!
035360  010137  001636                          MOV     R1,$TMP2
035364  062737  177776  001636                  ADD     #-2,$TMP2
035372  012737  000000  001634                  MOV     #0,$TMP1
035400  104001                                  ERROR   1




035402  020537  035544          MM27:           CMP     R5,MMPAT2               ;IS PATTERN CORRECT?
035406  001402                                  BEQ     MM28
035410  004737  035626                          JSR     PC,MMERR4


035414                          MM28:

035414  012205                                  MOV     (R2)+,R5        ;CHECK THE COMPLIMENTED

035416  005762  177776                          TST     -2(R2)

                                                                        ;SEE IF REFERENCE ADDRESS
035422  032737  000010  177752                  BIT     #10,@#HITMIS    ;IS A HIT.
035430  001011                                  BNE     MM29
                                                                        ;IF NOT ERROR!
```

```
035432  010237  001636                    MOV     R2,$TMP2
035436  062737  177776  001636            ADD     #-2,$TMP2
035444  012737  000001  001634            MOV     #1,$TMP1
035452  104001                            ERROR   1




035454  020537  035546           MM29:    CMP     R5,MMPAT3              ;IS PATTERN CORRECT?
035460  001402                            BEQ     MM30
035462  004737  035646                    JSR     PC,MMERR5


035466                           MM30:

035466  012205                            MOV     (R2)+,R5       ;CHECK THE COMPLIMENTED

035470  005762  177776                    TST     -2(R2)

                                                                 ;SEE IF REFERENCE ADDRESS
035474  032737  000010  177752            BIT     #10,@#HITMIS   ;IS A HIT.
035502  001011                            BNE     MM31
                                                                 ;IF NOT ERROR
035504  010237  001636                    MOV     R2,$TMP2
035510  062737  177776  001636            ADD     #-2,$TMP2
035516  012737  000001  001634            MOV     #1,$TMP1
035524  104001                            ERROR   1




035526  020537  035550           MM31:    CMP     R5,MMPAT4              ;IS PATTERN CORRECT?
035532  001464                            BEQ     MM32
035534  004737  035646                    JSR     PC,MMERR5

035540  000461                            BR      MM32           ;FINISHED THIS TEST.

035542  000000                   MMPAT1:  .WORD   0              ;THIS IS THE TEST PATTERN
035544  000000                   MMPAT2:  .WORD   0              ;TABLE.
035546  000000                   MMPAT3:  .WORD   0
035550  000000                   MMPAT4:  .WORD   0

035552  000000                   MMTMP1:  .WORD   0              ;THIS AREA IS USED TO ESTABLISH
035554  000004                   MMTMP2:  .BLKW   4              ;A TEST LOCATION WHOSE HITS WON'T
                                                                 ;BE INTERFERRED WITH BY THE CODE
                                                                 ;IN THE REST OF THIS TEST.

035564  005037  001634           MMERR1:  CLR     $TMP1          ;COME HERE TO REPORT
035570  010137  001642                    MOV     R1,$TMP4       ;GROUP 0 ERROR,WHILE READING
035574  000405                            BR      MMERR3         ;A BYTE INTO R5

035576  012737  000001  001634   MMERR2:  MOV     #1,$TMP1               ;COME HERE TO REPORT
035604  010237  001642                    MOV     R2,$TMP4       ;GROUP 1 ERROR, READING A
                                                                 ;BYTE INTO R5.
035610  012637  001636           MMERR3:  MOV     (SP)+,$TMP2
035614  010537  001640                    MOV     R5,$TMP3
```

```

035620  104017                                 ERROR   17
035622  000177  144010                         JMP     @$TMP2

035626  005037  001634                MMERR4:  CLR     $TMP1            ;REPORT AN ERROR IN GROUP
035632  010137  001642                         MOV     R1,$TMP4         ;0 WHILE READING A WORD
035636  062737  177776  001642                 ADD     #-2,$TMP4
035644  000410                                 BR      MMERR6

035646  012737  000001  001634        MMERR5:  MOV     #1,$TMP1
035654  010237  001642                         MOV     R2,$TMP4
035660  062737  177776  001642                 ADD     #-2,$TMP4

035666  012637  001636                MMERR6:  MOV     (SP)+,$TMP2
035672  010537  001640                         MOV     R5,$TMP3

035676  104020                                 ERROR   20
035700  000177  143732                         JMP     @$TMP2

035704                                MM32:                             ;DONE!




                                      ;;*************************************************************
                                      ;*TEST 35       CACHE ARBITRATION AND HIGH SPEED I/O TEST
                                      ;*
                                      ;*THIS IS A TEST OF:
                                      ;*       1.      CACHE ARBITRATION
                                      ;*       2.      THE MASS BUS AND UNIBUS PORTS TO THE CACHE
                                      ;*       3.      HIGH SPEED I/O THROUGH THE CACHF
                                      ;*
                                      ;*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 THF 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 THRUUGH
                                      ;*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.
                                        ;*
                                        ;*
                                        ;;*****************************************************************
035704  000004                          TST35:  SCOPE
                                                                        ;SET THE SKAD REGISTER
035706  012737  042352  055572                  MOV     #TST36,SKAD     ;IN CASE THE TEST ABORTS.

035714  104414                                  RSET
035716  113737  001502  001632                  MOVB    $TSTNM,$TMPO

035724  012700  172340                          MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
035730  012701  077406                          MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
035734  012702  172300                          MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
035740  012703  000010                          MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
035744  010122                          64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
035746  077302                                  SOB     R3,64$          ;ADDRESSES.
035750  005020                                  CLR     (R0)+
035752  012720  000200                          MOV     #200,(R0)+
035756  012720  000400                          MOV     #400,(R0)+
035762  012720  000600                          MOV     #600,(R0)+
035766  012720  001000                          MOV     #1000,(R0)+
035772  012720  001200                          MOV     #1200,(R0)+
035776  012720  001400                          MOV     #1400,(R0)+
036002  012710  177600                          MOV     #177600,(R0)

036006  012737  000001  177572                  MOV     #1,@#MMR0
036014  012737  000060  172516                  MOV     #60,@#MMR3

036022  004737  042116                  INTO:   JSR     PC,GTBINT               ;INITIALIZE THE MEMORY BUFFER
                                                                                ;ALLOCATION ROUTINES.
036026  004737  057066                          JSR     PC,SIZDEV               ;GO DETERMINE WHAT DEVICES ARE
                                                                                ;PRESENT.
036032  005046                                  CLR     -(SP)                   ;MAKE THE WAIT LOOP ACCESSABLE
```

```
036034  012746  036304                   MOV     #WAITLP,-(SP)           ;TO AN 'RTI'.

036040  012700  057460          INT1:    MOV     #RS4DFL,R0              ;GET READY TO SEE WHAT DEVICES
036044  012701  036234                   MOV     #RS4CR,R1               ;ARE TO BE USED.
036050  012702  036246                   MOV     #RS4SUN,R2
036054  012703  036260                   MOV     #RS4ASS,R3
036060  012704  000005                   MOV     #5,R4

036064  005011                  INT2:    CLR     (R1)                    ;CLEAR THE UNIT NUMBER.
036066  005012                           CLR     (R2)                    ;CLEAR THE COUNTER.
036070  105710                           TSTB    (R0)                    ;IS THERE A DRIVE.
036072  001447                           BEQ     INT6                    ;BRANCH IF NOT.

036074  111005                           MOVB    (R0),R5                 ;OTHERWISE DETERMINE A UNIT NUM.
036076  104412                           SAVREG
036100  012700  000010                   MOV     #10,R0
036104  005001                           CLR     R1
036106  012702  000001                   MOV     #1,R2
036112  030205                  INT3:    BIT     R2,R5
036114  001405                           BEQ     INT4
036116  010137  036230                   MOV     R1,INTMP1
036122  104413                           RESREG
036124  000137  036144                   JMP     INT5
036130  005201                  INT4:    INC     R1
036132  006302                           ASL     R2
036134  077012                           SOB     R0,INT3
036136  104413                           RESREG
036140  000137  036212                   JMP     INT6

036144  013711  036232          INT5:    MOV     CYCNT,(R1)              ;FOUND THE DRIVE SO SET UP THE
036150  020127  036234                   CMP     R1,#RS4CR
036154  001001                           BNE     1$
036156  006311                           ASL     (R1)
036160  020127  036236          1$:      CMP     R1,#RP4CR
036164  001001                           BNE     2$
036166  006311                           ASL     (R1)
036170  020127  036240          2$:      CMP     R1,#RH4CR
036174  001001                           BNE     3$
036176  006311                           ASL     (R1)
036200  012746  000340          3$:      MOV     #340,-(SP)              ;PASS COUNT AND MAKE THE DRIVER
036204  011346                           MOV     (R3),-(SP)              ;ACCESSIBLE BY A 'RTI'.
036206  013712  036230                   MOV     INTMP1,(R2)

036212  005200                  INT6:    INC     R0
036214  005721                           TST     (R1)+           ;MOVE THE POINTERS TO THE NEXT DEVICE.
036216  022223                           CMP     (R2)+,(R3)+
036220  000240                           NOP
036222  077460                           SOB     R4,INT2


036224  000240                           NOP
036226  000002                           RTI                             ;START THE TEST!


                                ;THESE ARE SOME TABLES THAT ARE USED TO CONTROL AND SET UP THIS TEST.
036230  000000                  INTMP1: .WORD   0
```

```


 036232  000010                          CYCNT:  .WORD   10                         ;THE PASS COUNT!!!!

 036234  000000                          RS4CR:  .WORD   0                          ;PASS COUNT FOR EACH DEVICE.
 036236  000000                          RP4CR:  .WORD   0
 036240  000000                          RH4CR:  .WORD   0
 036242  000000                          RK5CR:  .WORD   0
 036244  000000                          UBECR:  .WORD   0

 036246  000000                          RS4SUN: .WORD   0                          ;THE DRIVE NUMBER USED FOR EACH
 036250  000000                          RP4SUN: .WORD   0                          ;DEVICE.
 036252  000000                          RH4SUN: .WORD   0
 036254  000000                          RK5SUN: .WORD   0
 036256  000000                          UBESUN: .WORD   0

         036260                          SETBLE=RS4ASS
 036260  036350                          RS4ASS: .WORD   DRRS4                      ;STARTING ADDRESSES OF EACH DRIVER.
 036262  037162                          RP4ASS: .WORD   DRRP4
 036264  037774                          RH4ASS: .WORD   DRRH4
 036266  040566                          RK5ASS: .WORD   DRRK5
 036270  041400                          UBEASS: .WORD   DRUBE

 036272  000000                          RS4RB:  .WORD   0                          ;WRITE AND READ BUFFERS OF EACH DEVICE.
 036274  000000                          RP4RB:  .WORD   0
 036276  000000                          RH4RB:  .WORD   0
 036300  000000                          RK5RB:  .WORD   0
 036302  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.
 036304  000230                          WAITLP: SPL     0               ;LOWER THE PRIORITY.
 036306  005737  036242                          TST     RK5CR           ;WAIT FOR INTERRUPT OR ZERO PASS COUNT.
 036312  001374                                  BNE     WAITLP
 036314  005737  036244                          TST     UBECR
 036320  001371                                  BNE     WAITLP
 036322  005737  036236                          TST     RP4CR
 036326  001366                                  BNE     WAITLP
 036330  005737  036234                          TST     RS4CR
 036334  001363                                  BNE     WAITLP
 036336  005737  036240                          TST     RH4CR
 036342  001360                                  BNE     WAITLP

 036344  000137  042350                          JMP     INDONE          ;FINISHED!!!



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

 036350  000240                          DRRS4:  NOP
 036352  012737  007007  037156                  MOV     #7007,DRS4T1    ;INITIALIZE THE RANDOM DISK ADDRESS
 036360  012737  006006  037160                  MOV     #6006,DRS4T2    ;GENERATER.
 036366  012737  005005  036632                  MOV     #5005,RS4AA3

```

```
036374  000240                          RS4AA:  NOP
036376  000240                                  NOP
036400  104412                                  SAVREG
036402  004737  042232                          JSR     PC,GETBUF       ;GET A MEMORY BUFFER.
036406  036272                                  .WORD   RS4RB
036410  013701  036272                          MOV     RS4RB,R1
036414  005000                                  CLR     R0
036416  073027  000014                          ASHC    #12.,R0

036422  000237                                  SPL     7               ;GET A RANDOM DISK ADDRESS.
036424  013737  037156  054370                  MOV     DRS4T1,$HINUM
036432  013737  037160  054372                  MOV     DRS4T2,$LONUM
036440  004737  054272                          JSR     PC,$RAND
036444  013737  054370  037156                  MOV     $HINUM,DRS4T1
036452  013737  054372  03716C                  MOV     $LONUM,DRS4T2
036460  000230                                  SPL     0

036462  013702  036246                          MOV     RS4SUN,R2               ;SET UP THE DEVICE UNIT NUM.
036466  110237  037027                          MOVB    R2,RS4112
036472  110237  036655                          MOVB    R2,RS4BB
036476  110237  036721                          MOVB    R2,RS4HH
036502  110237  036765                          MOVB    R2,RS4NN

036506  013703  037156                          MOV     DRS4T1,R3               ;SET UP THE DISK ADDRESS.
036512  013704  037160                          MOV     DRS4T2,R4
036516  010337  036656                          MOV     R3,RS4CC
036522  010337  037030                          MOV     R3,RS4113
036526  010337  036722                          MOV     R3,RS4II
036532  010337  036766                          MOV     R3,RS4OO
036536  010437  036660                          MOV     R4,RS4DD
036542  010437  036724                          MOV     R4,RS4JJ
036546  010437  037032                          MOV     R4,RS4114
036552  010437  036770                          MOV     R4,RS4PP

036556  010137  036634                          MOV     R1,RS4AA1               ;SET THE MEMORY ADDRESS.
036562  010137  036662                          MOV     R1,RS4EE
036566  010137  036726                          MOV     R1,RS4KK
036572  010137  036772                          MOV     R1,RS4QQ
036576  010137  037034                          MOV     R1,RS4115
036602  010037  036774                          MOV     R0,RS4RR
036606  010037  037036                          MOV     R0,RS4116
036612  010037  036636                          MOV     R0,RS4AA2
036616  010037  036664                          MOV     R0,RS4FF
036622  010037  036730                          MOV     R0,RS4LL

036626  104413                                  RESREG

036630  104425                                  WRRAND                  ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
036632  000000                          RS4AA3: .WORD   0
036634  000000                          RS4AA1: .WORD   0
036636  000000                          RS4AA2: .WORD   0
036640  004000                                  .WORD   4000
036642  005237  036632                          INC     RS4AA3

036646  000240                                  NOP
036650  000237                                  SPL     7
```

```
036652  104426                   CALRS4          ;GET THE RS4 TO DO THE TRANSFER FROM MEMORY
036654     161                   .BYTE   161
036655     000           RS4BB:  .BYTE   0
036656  000000           RS4CC:  .WORD   0
036660  000000           RS4DD:  .WORD   0
036662  000000           RS4EE:  .WORD   0
036664  000000           RS4FF:  .WORD   0
036666  004000                   .WORD   4000
036670  036706                   .WORD   RS4GG

036672  000240                   NOP
036674  004737  037104           JSR     PC,RS4YY
036700  005066  000002           CLR     2(SP)
036704  000002                   RTI                     ;GO DO SOMETHING ELSE WHILE WAITING
                                                         ;FOR THE INTERRUPT!

036706  000240           RS4GG:  NOP
036710  004737  037104           JSR     PC,RS4YY                ;SEE IF THERE WERE ANY ERRORS.

036714  000237                   SPL     7
036716  104426                   CALRS4          ;DO THE WRITE CHECK
036720     151                   .BYTE   151
036721     000           RS4HH:  .BYTE   0
036722  000000           RS4II:  .WORD   0
036724  000000           RS4JJ:  .WORD   0
036726  000000           RS4KK:  .WORD   0
036730  000000           RS4LL:  .WORD   0
036732  004000                   .WORD   4000
036734  036752                   .WORD   RS4MM

036736  000240                   NOP
036740  004737  037104           JSR     PC,RS4YY
036744  005066  000002           CLR     2(SP)
036750  000002                   RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.

036752  000240           RS4MM:  NOP
036754  004737  037104           JSR     PC,RS4YY        ;SEE IF THERE WERE ANY ERRORS.


036760  000237                   SPL     7
036762  104426                   CALRS4          ;READ THE DISK.
036764     171                   .BYTE   171
036765     000           RS4NN:  .BYTE   0
036766  000000           RS4OO:  .WORD   0
036770  000000           RS4PP:  .WORD   0
036772  000000           RS4QQ:  .WORD   0
036774  000000           RS4RR:  .WORD   0
036776  004000                   .WORD   4000
037000  037016                   .WORD   RS4111

037002  000240                   NOP
037004  004737  037104           JSR     PC,RS4YY
037010  005066  000002           CLR     2(SP)
037014  000002                   RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.

037016  004737  037104   RS4111: JSR     PC,RS4YY
```

```
037022  000237                                  SPL     7

037024  104426                                  CALRS4
037026     151                                  .BYTE   151
037027     000                         RS4112:  .BYTE   0
037030  000000                         RS4113:  .WORD   0
037032  000000                         RS4114:  .WORD   0
037034  000000                         RS4115:  .WORD   0
037036  000000                         RS4116:  .WORD   0
037040  004000                                  .WORD   4000
037042  037060                                  .WORD   RS4SS
037044  000240                                  NOP
037046  004737  037104                          JSR     PC,RS4YY
037052  005066  000002                          CLR     2(SP)
037056  000002                                  RTI

037060  000240                         RS4SS:   NOP
037062  004737  037104                          JSR     PC,RS4YY        ;SEE IF ANY ERRORS OCCURRED.

037066  005337  036234                          DEC     RS4CR           ;DECRIMENT THE PASS COUNT.
037072  001001                                  BNE     RS4XX           ;IF NOT DONE CONTINUE.
037074  000002                                  RTI                     ;IF DONE GET OUT.

037076  000240                         RS4XX:   NOP
037100  000137  036374                          JMP     RS4AA           ;RESTART.

037104  000240                         RS4YY:   NOP
037106  005737  061004                          TST     RS4ER1   SEE IF ANY ERRORS OCCURRED.
037112  001420                                  BEQ     RS4ZZ           ;IF NOT THEN RETURN TO CALL.

037114  000237                                  SPL     7
037116  005037  036234                          CLR     RS4CR           ;IF YES THEN CLEAR THE PASS COUNT.
037122  013737  061006  001634                  MOV     RS4ER2,$TMP1    ;AND MAKE AN ERROR CALL.
037130  013737  061012  001640                  MOV     RS4ER4,$TMP3
037136  013737  061010  001636                  MOV     RS4ER3,$TMP2
037144  104154                                  ERROR   154
037146  000230                                  SPL     0
037150  005726                                  TST     (SP)+
037152  000002                                  RTI                     ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                                        ;FROM THE TEST.

037154  000207                         RS4ZZ:   RTS     PC              ;THERE WERE NO ERRORS.

037156  000000                         DRS4T1:  .WORD   0
037160  000000                         DRS4T2:  .WORD   0



                                       ;THIS IS THE RP4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                       ;TEST.

037162  000240                         DRRP4:   NOP
037164  012737  004004  037770                  MOV     #4004,DRP4T1    ;INITIALIZE THE RANDOM DISK ADDRESS
037172  012737  003003  037772                  MOV     #3003,DRP4T2    ;GENERATER.
037200  012737  002002  037444                  MOV     #2002,RP4AA3

```

```
037206  000240                          RP4AA:  NOP
037210  000240                                  NOP
037212  104412                                  SAVREG
037214  004737  042232                          JSR     PC,GETBUF         ;GET A MEMORY BUFFER.
037220  036274                                  .WORD   RP4RB
037222  013701  036274                          MOV     RP4RB,R1
037226  005000                                  CLR     R0
037230  073027  000014                          ASHC    #12.,R0

037234  000237                                  SPL     7                 ;GET A RANDOM DISK ADDRESS.
037236  013737  037770  054370                  MOV     DRP4T1,$HINUM
037244  013737  037772  054372                  MOV     DRP4T2,$LONUM
037252  004737  054272                          JSR     PC,$RAND
037256  013737  054370  037770                  MOV     $HINUM,DRP4T1
037264  013737  054372  037772                  MOV     $LONUM,DRP4T2
037272  000230                                  SPL     0

037274  013702  036250                          MOV     RP4SUN,R2                 ;SET UP THE DEVICE UNIT NUM.
037300  110237  037641                          MOVB    R2,RP4112
037304  110237  037467                          MOVB    R2,RP4BB
037310  110237  037533                          MOVB    R2,RP4HH
037314  110237  037577                          MOVB    R2,RP4NN

037320  013703  037770                          MOV     DRP4T1,R3                 ;SET UP THE DISK ADDRESS.
037324  013704  037772                          MOV     DRP4T2,R4
037330  010337  037470                          MOV     R3,RP4CC
037334  010337  037642                          MOV     R3,RP4113
037340  010337  037534                          MOV     R3,RP4II
037344  010337  037600                          MOV     R3,RP4OO
037350  010437  037472                          MOV     R4,RP4DD
037354  010437  037536                          MOV     R4,RP4JJ
037360  010437  037644                          MOV     R4,RP4114
037364  010437  037602                          MOV     R4,RP4PP

037370  010137  037446                          MOV     R1,RP4AA1                 ;SET THE MEMORY ADDRESS.
037374  010137  037474                          MOV     R1,RP4EE
037400  010137  037540                          MOV     R1,RP4KK
037404  010137  037604                          MOV     R1,RP4QQ
037410  010137  037646                          MOV     R1,RP4115
037414  010037  037606                          MOV     R0,RP4RR
037420  010037  037650                          MOV     R0,RP4116
037424  010037  037450                          MOV     R0,RP4AA2
037430  010037  037476                          MOV     R0,RP4FF
037434  010037  037542                          MOV     R0,RP4LL

037440  104413                                  RESREG

037442  104425                                  WRRAND                    ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
037444  000000                          RP4AA3: .WORD   0
037446  000000                          RP4AA1: .WORD   0
037450  000000                          RP4AA2: .WORD   0
037452  004000                                  .WORD   4000
037454  005237  037444                          INC     RP4AA3

037460  000240                                  NOP
037462  000237                                  SPL     7
```

```
037464  104427                          CALRP4                  ;GET THE RP4 TO DO THE TRANSFER FROM MEMORY
037466     161                          .BYTE   161
037467     000                  RP4BB:  .BYTE   0
037470  000000                  RP4CC:  .WORD   0
037472  000000                  RP4DD:  .WORD   0
037474  000000                  RP4EE:  .WORD   0
037476  000000                  RP4FF:  .WORD   0
037500  004000                          .WORD   4000
037502  037520                          .WORD   RP4GG

037504  000240                          NOP
037506  004737  037716                  JSR     PC,RP4YY
037512  005066  000002                  CLR     2(SP)
037516  000002                          RTI                             ;GO DO SOMETHING ELSE WHILE WAITING
                                                                        ;FOR THE INTERRUPT!

037520  000240                  RP4GG:  NOP
037522  004737  037716                  JSR     PC,RP4YY                        ;SEE IF THERE WERE ANY ERRORS.

037526  000237                          SPL     7
037530  104427                          CALRP4                  ;DO THE WRITE CHECK
037532     151                          .BYTE   151
037533     000                  RP4HH:  .BYTE   0
037534  000000                  RP4II:  .WORD   0
037536  000000                  RP4JJ:  .WORD   0
037540  000000                  RP4KK:  .WORD   0
037542  000000                  RP4LL:  .WORD   0
037544  004000                          .WORD   4000
037546  037564                          .WORD   RP4MM

037550  000240                          NOP
037552  004737  037716                  JSR     PC,RP4YY
037556  00506   000002                  CLR     2(SP)
037562  00000                           RTI                             ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.

037564  000240                  RP4MM:  NOP
037566  004737  037716                  JSR     PC,RP4YY                ;SEE IF THERE WERE ANY ERRORS.


037572  000237                          SPL     7
037574  104427                          CALRP4                  ;READ THE DISK.
037576     171                          .BYTE   171
037577     000                  RP4NN:  .BYTE   0
037600  000000                  RP4OO:  .WORD   0
037602  000000                  RP4PP:  .WORD   0
037604  000000                  RP4QQ:  .WORD   0
037606  000000                  RP4RR:  .WORD   0
037610  004000                          .WORD   4000
037612  037630                          .WORD   RP4111

037614  000240                          NOP
037616  004737  037716                  JSR     PC,RP4YY
037622  005066  000002                  CLR     2(SP)
037626  000002                          RTI                     ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.

037630  004737  037716          RP4111: JSR     PC,RP4YY
```

```
037634  000237                                    SPL     7

037636  104427                                    CALRP4
037640     151                                    .BYTE   151
037641     000                          RP4112:   .BYTE   0
037642  000000                          RP4113:   .WORD   0
037644  000000                          RP4114:   .WORD   0
037646  000000                          RP4115:   .WORD   0
037650  000000                          RP4116:   .WORD   0
037652  004000                                    .WORD   4000
037654  037672                                    .WORD   RP4SS
037656  000240                                    NOP
037660  004737  037716                            JSR     PC,RP4YY
037664  005066  000002                            CLR     2(SP)
037670  000002                                    RTI

037672  000240                          RP4SS:    NOP
037674  004737  037716                            JSR     PC,RP4YY          ;SEE IF ANY ERRORS OCCURRED.

037700  005337  036236                            DEC     RP4CR             ;DECRIMENT THE PASS COUNT.
037704  001001                                    BNE     RP4XX             ;IF NOT DONE CONTINUE.
037706  000002                                    RTI                       ;IF DONE GET OUT.

037710  000240                          RP4XX:    NOP
037712  000137  037206                            JMP     RP4AA             ;RESTART.

037716  000240                          RP4YY:    NOP
037720  005737  060034                            TST     RP4ER1  ;SEE IF ANY ERRORS OCCURRED.
037724  001420                                    BEQ     RP4ZZ             ;IF NOT THEN RETURN TO CALL.

037726  000237                                    SPL     7
037730  005037  036236                            CLR     RP4CR             ;IF YES THEN CLEAR THE PASS COUNT.
037734  013737  060036  001634                    MOV     RP4ER2,$TMP1      ;AND MAKE AN ERROR CALL.
037742  013737  060042  001640                    MOV     RP4ER4,$TMP3
037750  013737  060040  001636                    MOV     RP4ER3,$TMP2
037756  104155                                    ERROR   155
037760  000230                                    SPL     0
037762  005726                                    TST     (SP)+
037764  000002                                    RTI                       ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                                            ;FROM THE TEST.

037766  000207                          RP4ZZ:    RTS     PC                ;THERE WERE NO ERRORS.

037770  000000                          DRP4T1:   .WORD   0
037772  000000                          DRP4T2:   .WORD   0



                                        ;THIS IS THE RH4 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                        ;TEST.

037774  000240                          DRRH4:    NOP
037776  012737  070070  040562                    MOV     #70070,DRH4T1     ;INITIALIZE THE RANDOM DISK ADDRESS
040004  012737  060060  040564                    MOV     #60060,DRH4T2     ;GENERATER.
040012  012737  050050  040236                    MOV     #50050,RH4AA3

```

```
040020  000240                          RH4AA:   NOP
040022  000240                                   NOP
040024  104412                                   SAVREG
040026  004737  042232                           JSR     PC,GETBUF          ;GET A MEMORY BUFFER.
040032  036276                                   .WORD   RH4RB
040034  013701  036276                           MOV     RH4RB,R1
040040  005000                                   CLR     R0
040042  073027  000014                           ASHC    #12.,R0

040046  000237                                   SPL     7                  ;GET A RANDOM DISK ADDRESS.
040050  013737  040562  054370                   MOV     DRH4T1,$HINUM
040056  013737  040564  054372                   MOV     DRH4T2,$LONUM
040064  004737  054272                           JSR     PC,$RAND
040070  013737  054370  040562                   MOV     $HINUM,DRH4T1
040076  013737  054372  040564                   MOV     $LONUM,DRH4T2
040104  000230                                   SPL     0

040106  013702  036252                           MOV     RH4SUN,R2                 ;SET UP THE DEVICE UNIT NUM.
040112  110237  040433                           MOVB    R2,RH4112
040116  110237  040261                           MOVB    R2,RH4BB
040122  110237  040325                           MOVB    R2,RH4HH
040126  110237  040371                           MOVB    R2,RH4NN

040132  013703  040562                           MOV     DRH4T1,R3                 ;SET UP THE DISK ADDRESS.
040136  013704  040564                           MOV     DRH4T2,R4
040142  010337  040262                           MOV     R3,RH4CC
040146  010337  040434                           MOV     R3,RH4113
040152  010337  040326                           MOV     R3,RH4II
040156  010337  040372                           MOV     R3,RH4OO

040162  010137  040240                           MOV     R1,RH4AA1                 ;SET THF MEMORY ADDRESS.
040166  010137  040266                           MOV     R1,RH4EE
040172  010137  040332                           MOV     R1,RH4KK
040176  010137  040376                           MOV     R1,RH4QQ
040202  010137  040440                           MOV     R1,RH4115
040206  010037  040400                           MOV     R0,RH4RR
040212  010037  040442                           MOV     R0,RH4116
040216  010037  040242                           MOV     R0,RH4AA2
040222  010037  040270                           MOV     R0,RH4FF
040226  010037  040334                           MOV     R0,RH4LL

040232  104413                                   RESREG

040234  104425                                   WRRAND                     ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
040236  000000                          RH4AA3:  .WORD   0
040240  000000                          RH4AA1:  .WORD   0
040242  000000                          RH4AA2:  .WORD   0
040244  004000                                   .WORD   4000
040246  005237  040236                           INC     RH4AA3

040252  000240                                   NOP
040254  000237                                   SPL     7
040256  104430                                   CALRH4           ;GET THE RH4 TO DO THE TRANSFER FROM MEMORY
040260     161                                   .BYTE   161
040261     000                          RH4BB:   .BYTE   0
040262  000000                          RH4CC:   .WORD   0
```

```
 7293  040264  000000                   RH4DD:  .WORD   0
 7294  040266  000000                   RH4EE:  .WORD   0
 7295  040270  000000                   RH4FF:  .WORD   0
 7296  040272  004000                           .WORD   4000
 7297  040274  040312                           .WORD   RH4GG
 7298
 7299  040276  000240                           NOP
 7300  040300  004737  040510                   JSR     PC,RH4YY
 7301  040304  005066  000002                   CLR     2(SP)
 7302  040310  000002                           RTI                         ;GO DO SOMETHING ELSE WHILE WAITING
 7303                                                                       ;FOR THE INTERRUPT.
 7304
 7305  040312  000240                   RH4GG:  NOP
 7306  040314  004737  040510                   JSR     PC,RH4YY                    ;SEE IF THERE WERE ANY ERRORS.
 7307
 7308  040320  000237                           SPL     7
 7309  040322  104430                           CALRH4          ;DO THE WRITE CHECK
 7310  040324     171                           .BYTE   171
 7311  040325     000                   RH4HH:  .BYTE   0
 7312  040326  000000                   RH4II:  .WORD   0
 7313  040330  000000                   RH4JJ:  .WORD   0
 7314  040332  000000                   RH4KK:  .WORD   0
 7315  040334  000000                   RH4LL:  .WORD   0
 7316  040336  004000                           .WORD   4000
 7317  040340  040356                           .WORD   RH4MM
 7318
 7319  040342  000240                           NOP
 7320  040344  004737  040510                   JSR     PC,RH4YY
 7321  040350  005066  000002                   CLR     2(SP)
 7322  040354  000002                           RTI                         ;DO SOMETHING ELSE WHILE WAITING FOR INTERRUPT.
 7323
 7324  040356  000240                   RH4MM:  NOP
 7325  040360  004737  040510                   JSR     PC,RH4YY            ;SEE IF THERE WERE ANY ERRORS.
 7326
 7327
 7328  040364  000237                           SPL     7
 7329  040366  104430                           CALRH4          ;READ THE DISK.
 7330  040370     151                           .BYTE   151
 7331  040371     000                   RH4NN:  .BYTE   0
 7332  040372  000000                   RH400:  .WORD   0
 7333  040374  000000                   RH4PP:  .WORD   0
 7334  040376  000000                   RH4QQ:  .WORD   0
 7335  040400  000000                   RH4RR:  .WORD   0
 7336  040402  004000                           .WORD   4000
 7337  040404  040422                           .WORD   RH4111
 7338
 7339  040406  000240                           NOP
 7340  040410  004737  040510                   JSR     PC,RH4YY
 7341  040414  005066  000002                   CLR     2(SP)
 7342  040420  000002                           RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.
 7343
 7344  040422  004737  040510           RH4111: JSR     PC,RH4YY
 7345  040426  000237                           SPL     7
 7346
 7347  040430  104430                           CALRH4
 7348  040432     171                           .BYTE   171
```

```
040433     000                      RH4112: .BYTE   0
040434  000000                      RH4113: .WORD   0
040436  000000                      RH4114: .WORD   0
040440  000000                      RH4115: .WORD   0
040442  000000                      RH4116: .WORD   0
040444  004000                              .WORD   4000
040446  040464                              .WORD   RH4SS
040450  000240                              NOP
040452  004737  040510                      JSR     PC,RH4YY
040456  005066  000002                      CLR     2(SP)
040462  000002                              RTI

040464  000240                      RH4SS:  NOP
040466  004737  040510                      JSR     PC,RH4YY        ;SEE IF ANY ERRORS OCCURRED.

040472  005337  036240                      DEC     RH4CR           ;DECRIMENT THE PASS COUNT.
040476  001001                              BNE     RH4XX           ;IF NOT DONE CONTINUE.
040500  000002                              RTI                     ;IF DONE GET OUT!

040502  000240                      RH4XX:  NOP
040504  000137  040020                      JMP     RH4AA           ;RESTART.

040510  000240                      RH4YY:  NOP
040512  005737  063500                      TST     RH4ER1  ;SEE IF ANY ERRORS OCCURRED.
040516  001420                              BEQ     RH4ZZ           ;IF NOT THEN RETURN TO CALL.

040520  000237                              SPL     7
040522  005037  036240                      CLR     RH4CR           ;IF YES THEN CLEAR THE PASS COUNT.
040526  013737  063502  001634              MOV     RH4ER2,$TMP1    ;AND MAKE AN ERROR CALL.
040534  013737  063506  001640              MOV     RH4ER4,$TMP3
040542  013737  063504  001636              MOV     RH4ER3,$TMP2
040550  104156                              ERROR   156
040552  000230                              SPL     0
040554  005726                              TST     (SP)+
040556  000002                              RTI                     ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                                    ;FROM THE TEST.

040560  000207                      RH4ZZ.  RTS     PC              ;THERE WERE NO ERRORS.

040562  000000                      DRH4T1: .WORD   0
040564  000000                      DRH4T2: .WORD   0




                                    ;THIS IS THE RK5 DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                                    ;TEST.

040566  000240                      DRRK5:  NOP
040570  012737  030030  041374              MOV     #30030,DRK5T1   ;INITIALIZE THE RANDOM DISK ADDRESS
040576  012737  040040  041376              MOV     #40040,DRK5T2   ;GENERATER.
040604  012737  050050  041050              MOV     #50050,RK5AA3

040612  000240                      RK5AA:  NOP
040614  000240                              NOP
040616  104412                              SAVREG
```

```
040620  004737  042232                     JSR     PC,GETBUF         ;GET A MEMORY BUFFER.
040624  036300                             .WORD   RK5RB
040626  013701  036300                     MOV     RK5RB,R1
040632  005000                             CLR     R0
040634  073027  000014                     ASHC    #12.,R0

040640  000237                             SPL     7                 ;GET A RANDOM DISK ADDRESS.
040642  013737  041374  054370             MOV     DRK5T1,$HINUM
040650  013737  041376  054372             MOV     DRK5T2,$LONUM
040656  004737  054272                     JSR     PC,$RAND
040662  013737  054370  041374             MOV     $HINUM,DRK5T1
040670  013737  054372  041376             MOV     $LONUM,DRK5T2
040676  000230                             SPL     0

040700  013702  036254                     MOV     RK5SUN,R2                   ;SET UP THE DEVICE UNIT NUM.
040704  110237  041245                     MOVB    R2,RK5112
040710  110237  041073                     MOVB    R2,RK5BB
040714  110237  041137                     MOVB    R2,RK5HH
040720  110237  041203                     MOVB    R2,RK5NN

040724  013703  041374                     MOV     DRK5T1,R3                   ;SET UP THE DISK ADDRESS.
040730  013704  041376                     MOV     DRK5T2,R4
040734  010337  041074                     MOV     R3,RK5CC
040740  010337  041246                     MOV     R3,RK5113
040744  010337  041140                     MOV     R3,RK5II
040750  010337  041204                     MOV     R3,RK5OO
040754  010437  041076                     MOV     R4,RK5DD
040760  010437  041142                     MOV     R4,RK5JJ
040764  010437  041250                     MOV     R4,RK5114
040770  010437  041206                     MOV     R4,RK5PP

040774  010137  041052                     MOV     R1,RK5AA1                   ;SET THE MEMORY ADDRESS.
041000  010137  041100                     MOV     R1,RK5EE
041004  010137  041144                     MOV     R1,RK5KK
041010  010137  041210                     MOV     R1,RK5QQ
041014  010137  041252                     MOV     R1,RK5115
041020  010037  041212                     MOV     R0,RK5RR
041024  010037  041254                     MOV     R0,RK5116
041030  010037  041054                     MOV     R0,RK5AA2
041034  010037  041102                     MOV     R0,RK5FF
041040  010037  041146                     MOV     R0,RK5LL

041044  104413                             RESREG

041046  104425                             WRRAND                    ;FILL THE MEMORY BUFFER WITH RANDOM DATA.
041050  000000                    RK5AA3:  .WORD   0
041052  000000                    RK5AA1:  .WORD   0
041054  000000                    RK5AA2:  .WORD   0
041056  004000                             .WORD   4000
041060  005237  041050                     INC     RK5AA3

041064  000240                             NOP
041066  000237                             SPL     7
041070  104431                             CALRK5          ;GET THE RK5 TO DO THE TRANSFER FROM MEMORY
041072     103                             .BYTE   103
041073     000                    RK5BB:   .BYTE   0
```

```
041074  000000                   RK5CC:  .WORD   0
041076  000000                   RK5DD:  .WORD   0
041100  000000                   RK5EE:  .WORD   0
041102  000000                   RK5FF:  .WORD   0
041104  004000                           .WORD   4000
041106  041124                           .WORD   RK5GG

041110  000240                           NOP
041112  004737  041322                   JSR     PC,RK5YY
041116  005066  000002                   CLR     2(SP)
041122  000002                           RTI                     ;GO DO SOMETHING ELSE WHILE WAITING
                                                                 ;FOR THE INTERRUPT!

041124  000240                   RK5GG:  NOP
041126  004737  041322                   JSR     PC,RK5YY                ;SEE IF THERE WERE ANY ERRORS.

041132  000237                           SPL     7
041134  104431                           CALRK5          ;DO THE WRITE CHECK
041136     107                           .BYTE   107
041137     000                   RK5HH:  .BYTE   0
041140  000000                   RK5II:  .WORD   0
041142  000000                   RK5JJ:  .WORD   0
041144  000000                   RK5KK:  .WORD   0
041146  000000                   RK5LL:  .WORD   0
041150  004000                           .WORD   4000
041152  041170                           .WORD   RK5MM

041154  000240                           NOP
041156  004737  041322                   JSR     PC,RK5YY
041162  005066  000002                   CLR     2(SP)
041166  000002                           RTI                     ;DO SOMETHING E SE WHILE WAITING FOR INTERRUPT.

041170  000240                   RK5MM:  NOP
041172  004737  041322                   JSR     PC,RK5YY        ;SEE IF THERE WERE ANY ERRORS.


041176  000237                           SPL     7
041200  104431                           CALRK5          ;READ THE DISK.
041202     105                           .BYTE   105
041203     000                   RK5NN.  .BYTE   0
041204  000000                   RK5OO:  .WORD   0
041206  000000                   RK5PP:  .WORD   0
041210  000000                   RK5QQ:  .WORD   0
041212  000000                   RK5RR:  .WORD   0
041214  004000                           .WORD   4000
041216  041234                           .WORD   RK5111

041220  000240                           NOP
041222  004737  041322                   JSR     PC,RK5YY
041226  005066  000002                   CLR     2(SP)
041232  000002                           RTI             ;DO SOMETHING ELSE WHILE WAITING FOR THE INTER.

041234  004737  041322           RK5111: JSR     PC,RK5YY
041240  000237                           SPL     7

041242  104431                           CALRK5
```

```
041244     107                        .BYTE   107
041245     000               RK5112:  .BYTE   0
041246  000000               RK5113:  .WORD   0
041250  000000               RK5114:  .WORD   0
041252  000000               RK5115:  .WORD   0
041254  000000               RK5116:  .WORD   0
041256  004000                        .WORD   4000
041260  041276                        .WORD   RK5SS
041262  000240                        NOP
041264  004737  041322                JSR     PC,RK5YY
041270  005066  000002                CLR     2(SP)
041274  000002                        RTI

041276  000240               RK5SS:   NOP
041300  004737  041322                JSR     PC,RK5YY         ;SEE IF ANY ERRORS OCCURRED.

041304  005337  036242                DEC     RK5CR            ;DECRIMENT THE PASS COUNT.
041310  001001                        BNE     RK5XX            ;IF NOT DONE CONTINUE.
041312  000002                        RTI                      ;IF DONE GET OUT!

041314  000240               RK5XX:   NOP
041316  000137  040612                JMP     RK5AA            ;RESTART.

041322  000240               RK5YY:   NOP
041324  005737  061740                TST     RK5ER1  ;SEE IF ANY ERRORS OCCURRED.
041330  001420                        BEQ     RK5ZZ            ;IF NOT THEN RETURN TO CALL.

041332  000237                        SPL     7
041334  005037  036242                CLR     RK5CR            ;IF YES THEN CLEAR THE PASS COUNT.
041340  013737  061742  001634        MOV     RK5ER2,$TMP1     ;AND MAKE AN ERROR CALL.
041346  013737  061746  001640        MOV     RK5ER4,$TMP3
041354  013737  061744  001636        MOV     RK5ER3,$TMP2
041362  104160                        ERROR   160
041364  000230                        SPL     0
041366  005726                        TST     (SP)+
041370  000002                        RTI                      ;RETURN TO WAIT LOOP, DROPPING THIS DEVICE
                                                               ;FROM THE TEST.

041372  000207               RK5ZZ:   RTS     PC               ;THERE WERE NO ERRORS.

041374  000000               DRK5T1:  .WORD   0
041376  000000               DRK5T2:  .WORD   0



                             ;THIS IS THE UBE DRIVER ROUTINE USED IN THE CACHE I/O ARBITRATION
                             ;TEST.
041400  012737  050050  041746 DRUBE:  MOV     #50050,DUBET1    ;INITIALIZE THE RANDOM DATA
041406  012737  060060  041750        MOV     #60060,DUBET2    ;GENERATER.
041414  012737  070070  041554        MOV     #70070,UBEAA3

041422  104412               UBEAA:   SAVREG
041424  004737  042232                JSR     PC,GETBUF        ;PICK UP A MEMORY BUFFER
041430  036302                        .WORD   UBERB

041432  013701  036302                MOV     UBERB,R1         ;COMPUTE THE MEMORY ADDRESS.
```

```
041436  005000                           CLR     R0
041440  073027  000014                   ASHC    #12.,R0
041444  010137  041556                   MOV     R1,UBEAA1
041450  010137  041602                   MOV     R1,UBEDD
041454  010137  041642                   MOV     R1,UBEII
041460  010037  041560                   MOV     R0,UBEAA2
041464  010037  041604                   MOV     R0,UBEEE
041470  010037  041644                   MOV     R0,UBEJJ

041474  000237                           SPL     7
041476  013737  041746  054370           MOV     DUBET1,$HINUM
041504  013737  041750  054372           MOV     DUBET2,$LONUM
041512  004737  054272                   JSR     PC,$RAND
041516  013737  054370  041746           MOV     $HINUM,DUBET1
041524  013737  054372  04175C           MOV     $LONUM,DUBET2
041532  000230                           SPL     0

041534  013703  041746                   MOV     DUBET1,R3          ;SET THE UNIBUS TESTER DATA REG.
041540  010337  041640                   MOV     R3,UBEHH
041544  010337  041600                   MOV     R3,UBECCC

041550  104413                           RESREG

041552  104425                           WRRAND                     ;FILL THE MEMORY BUFFER WITH
041554  000000                  UBEAA3:  .WORD   0                  ;RANDOM DATA.
041556  000000                  UBEAA1:  .WORD   0
041560  000000                  UBEAA2:  .WORD   0
041562  004000                           .WORD   4000
041564  005237  041554                   INC     UBEAA3

041570  000237                           SPL     7
041572  104432                           CALUBE                     ;DO A READ MEMORY FUNCTION.
041574  042543                           .WORD   42543
041576  000000                  UBEBB:   .WORD   0
041600  000000                  UBECCC:  .WORD   0
041602  000000                  UBEDD:   .WORD   0
041604  000000                  UBEEE:   .WORD   0
041606  010000                           .WORD   10000

041610  041624                           .WORD   UBEFF

041612  004737  041704                   JSR     PC,UBEYY
041616  005066  000002                   CLR     2(SP)
041622  000002                           RTI                        ;GO DO SOMETHING ELSE WHILE
                                                                    ;WAITING FOR INTERRUPT.
041624  004737  041704          UBEFF:   JSR     PC,UBEYY

041630  000237                           SPL     7
041632  104432                           CALUBE                     ;DO A WRITE MEMORY FUNCTION.
041634  042543                           .WORD   42543
041636  000000                  UBEGG:   .WORD   0
041640  000000                  UBEHH:   .WORD   0
041642  000000                  UBEII:   .WORD   0
041644  000000                  UBEJJ:   .WORD   0
041646  010000                           .WORD   10000
041650  041664                           .WORD   UBEKK
```

```

041652  004737  041704                  JSR     PC,UBEYY
041656  005066  000002                  CLR     2(SP)
041662  000002                          RTI                             ;GO DO SOMETHING ELSE WHILE
                                                                        ;WAITING FOR THE INTERRUPT.
041664  004737  041704          UBEKK:  JSR     PC,UBEYY

041670  005337  036244                  DEC     UBECR                   ;DECREMENT THE PASS COUNT.
041674  001001                          BNE     UBELL                   ;BR IF NOT DONE

041676  000002                          RTI                             ;IF DONE RETURN.
041700  000137  041422          UBELL:  JMP     UBEAA                   ;IF NOT DONE DO ANOTHER PASS.

041704  005737  062754          UBEYY:  TST     UBEER1                  ;WERE THERE ANY ERRORS?
041710  001415                          BEQ     UBEZZ                   ;BR IF NO.

041712  000237                          SPL     7                       ;IF THERE WERE REPORT DEVICE FAILURE.
041714  005037  036244                  CLR     UBECR
041720  013737  062756  001634          MOV     UBEER2,$TMP1
041726  013737  062760  001636          MOV     UBEER3,$TMP2
041734  104161                          ERROR   161
041736  005726                          TST     (SP)+
041740  000230                          SPL     0
041742  000002                          RTI                             ;RETURN WITH THIS DRIVER LOCKED OUT.
041744  000207                  UBEZZ:  RTS     PC                      ;NO ERRORS CONTINUE.

041746  000000                  DUBET1: .WORD   0
041750  000000                  DUBET2: .WORD   0




                                ;THIS ROUTINE IS USED TO GENERATE A BUFFER FULL OF RANDOM DATA.
                                ;IT IS CALLED USING THE TRAP TABLE CALL:
                                ;       WRRAND
                                ;       .WORD   HIGHNUM
                                ;       .WORD   LOADRS
                                ;       .WORD   HIGHADRS
                                ;       .WORD   WORDCOUNT
                                ;RET:
                                ;WHERE HIGHNUM IS THE HIGH ORDER PART OF THE NUMBER USED TP PRIME THE
                                ;RANDOM NUMBER GENERATER. THE LOW ORDER PART OF THAT NUMBER IS ASSUMED
                                ;TO BE ZERO. LOADRS AND HIGHADRS IS THE 22 BIT ADDRESS OF THE BUFFER
                                ;IN MEMORY WHICH WILL BE FILLED. WORDCOUNT IS THE NUMBER OF LOCATIONS
                                ;TO BE WRITTEN.
041752  000237                  RANDWR: SPL     7
041754  011637  042114                  MOV     (SP),RANDTP
041760  062716  000010                  ADD     #10,(SP)
041764  104412                          SAVREG
041766  013700  042114                  MOV     RANDTP,R0
041772  012001                          MOV     (R0)+,R1
041774  012002                          MOV     (R0)+,R2
041776  012003                          MOV     (R0)+,R3
042000  012004                          MOV     (R0)+,R4
042002  010237  042112                  MOV     R2,RLWT
042006  010337  042110                  MOV     R3,RHWT
042012  010137  054370                  MOV     R1,$HINUM
```

```
042016  005037  054372                    CLR     $LONUM

042022  013702  042110            1$:     MOV     RHWT,R2         ;COMPUTE THE VIRTUAL ADDRESS OF THE BUFFER WORD.
042026  013703  042112                    MOV     RLWT,R3
042032  073227  177772                    ASHC    #-6,R2
042036  010337  172354                    MOV     R3,@#KIPAR6
042042  013702  042112                    MOV     RLWT,R2
042046  042702  177700                    BIC     #177700,R2
042052  062702  140000                    ADD     #140000,R2
042056  004737  054272                    JSR     PC,$RAND
042062  013712  054370                    MOV     $HINUM,(R2)
042066  062737  000002  042112            ADD     #2,RLWT
042074  005537  042110                    ADC     RHWT
042100  077430                            SOB     R4,1$

042102  000230                            SPL     0
042104  104413                            RESRFG
042106  000002                            RTI

042110  000000                    RHWT:   .WORD   0
042112  000000                    RLWT:   .WORD   0
042114  000000                    RANDTP: .WORD   0

                                  ;THIS ROUTINE IS USED TO INITIALIZE THE GET BUFFER ROUTINE.
042116  012700  036272            GTBINT: MOV     #RS4RB,R0       ;CLEAR ALL THE BUFFER POINTERS.
042122  012701  000005                    MOV     #5,R1

042126  005020                    1$:     CLR     (R0)+
042130  077102                            SOB     R1,1$
042132  104417                            SIZE                    ;COMPUTE THE SIZE OF MEMORY.
042134  000000                    GTBILO: .WORD   0
042136  000000                    GTBIHI: .WORD   0
042140  062737  000002  042134            ADD     #2,GTBILO
042146  005537  042136                    ADC     GTBIHI
042152  013700  042136                    MOV     GTBIHI,R0       ;COMPUTE THE 2K BLOCK SIZE OF MEMORY.
042156  013701  042134                    MOV     GTBILO,R1
042162  073027  177764                    ASHC    #-12.,R0
042166  010137  042220                    MOV     R1,GTMSIZ
042172  162701  000011                    SUB     #11,R1
042176  010137  042222                    MOV     R1,AVMBL
042202  012737  123456  042224            MOV     #123456,GTRNL
042210  012737  123456  042226            MOV     #123456,GTRNH
042216  000207                            RTS     PC

042220  000000                    GTMSIZ: .WORD   0
042222  000000                    AVMBL:  .WORD   0
042224  000000                    GTRNL:  .WORD   0
042226  000000                    GTRNH:  .WORD   0
042230  000000                    GETMP1: .WORD   0

                                  ;THIS ROUTINE IS CALLED TO ALLOCATE A MEMORY BUFFER OF 2K WORDS LENGTH.
                                  ;IT IS CALLED USING A JSR PC INSTRUCTION FOLLOWED BY THE TABLE ENTRY
                                  ;OF RS4RB TO BE UPDATED.
042232  000237                    GETBUF: SPL     7               ;LOCK OUT INTERRUPTS.
042234  011637  042230                    MOV     (SP),GETMP1
042240  062716  000002                    ADD     #2,(SP)         ;PICK UP A POINTER TO THE ARGUMENT
```

```
                                                                ;AND UPDATE THE RETURN ADDRESS.
042244  104412                                  SAVREG
042246  013737  042224  054372  1$:     MOV     GTRNL,$LONUM
042254  013737  042226  054370          MOV     GTRNH,$HINUM
042262  004737  054272                  JSR     PC,$RAND
042266  013737  054372  042224          MOV     $LONUM,GTRNL
042274  013701  054370                  MOV     $HINUM,R1
042300  010137  042226                  MOV     R1,GTRNH
042304  005000                          CLR     R0
042306  071037  042222                  DIV     AVMBL,R0

042312  012702  036272                  MOV     #RS4RB,R2       ;SEE IF THIS AREA IS ALREADY IN USE.
042316  012703  000005                  MOV     #5,R3
042322  062701  000011                  ADD     #11,R1

042326  020122                  2$:     CMP     R1,(R2)+
042330  001746                          BEQ     1$              ;IF IT IS THEN TRY AGAIN.
042332  077303                          SOB     R3,2$

042334  017704  177670                  MOV     @GETMP1,R4      ;OTHERWISE GIVE THIS BUFFER TO THE DRIVER.
042340  010114                          MOV     R1,(R4)
042342  104413                          RESREG
042344  000230                          SPL     0
042346  000207                          RTS     PC


042350  104414                  INDONE: RSET




                                ;;*****************************************************************
                                ;*TEST 36       MASS BUS WRITE HIT CYCLE, INVALIDATION TEST
                                ;*
                                ;*THIS IS A TEST OF CACHE INVALIDATION ON MASS BUS CYCLES WHICH ARE
                                ;*WRITE HITS IN THE CACHE. A GROUP OF LOCATIONS IS MADE HITS AND THEN A
                                ;*MASS BUS DEVICE IS CALLED UPON TO DO TRANSFERS, WRITES TO THOSE
                                ;*LOCATIONS. THOSE WRITES SHOULD THUS BE INVALIDATED.
                                ;*
                                ;;*****************************************************************
042352  000004                  TST36:  SCOPE
                                                                ;SET THE SKAD REGISTER
042354  012737  050102  055572          MOV     #KT,SKAD        ;IN CASE THE TEST ABORTS.

042362  104414                          RSET
042364  113737  001502  001632          MOVB    $TSTNM,$TMP0
042372  004737  057066                  JSR     PC,SIZDEV       ;DETERMINE WHAT DEVICES ARE AVAILABLE.
042376  113737  057460  043112          MOVB    RS4DFL,RS4FT
042404  113737  057461  043113          MOVB    RP4DFL,RP4FT
042412  113737  057462  043114          MOVB    RH4DFL,RH4FT

042420  000137  043230          NN1:    JMP     NNDEV           ;GO COMPUTE THE DRIVE NUMBERS.

042424  005037  043110          NN2:    CLR     NNGRPF          ;FLAG WHICH DESIGNATES WHICH GOUP IS BEING
042430  012737  000044  043106          MOV     #S1M0,NNGRM     ;TESTED ON THIS PASS.
042436  012737  000030  043104          MOV     #S0M1,NNGRS     ;TEST GROUP ZERO FIRST.
```

```

042444  004737  043120          NN3:    JSR     PC,NNSTUP       ;GO MAKE THE TEST ADDRESSES HITS
042450  004777  000426                  JSR     PC,@NNUD                ;USE THE FIRST DEVICE.


042454  012700  140000                  MOV     #TESTR1,R0
042460  012701  000400                  MOV     #256.,R1        ;MAKE SURE THOSE ADDRESSES ARE MISSES.

042464  005710                  1$:     TST     (R0)
042466  032737  000010  177752          BIT     #10,@#HITMIS
042474  001430                          BEQ     2$

042476  013737  043110  001634          MOV     NNGRPF,$TMP1    ;GOT A HIT REPORT FAILURE.
042504  010037  001636                  MOV     R0,$TMP2
042510  005037  001640                  CLR     $TMP3
042514  023727  043102  042716          CMP     NNUD,#NNRS4     ;WAS THE RS4 DOING THE TRANSFER?
042522  001003                          BNE     11$             ;BRANCH IF NOT.
042524  104151                          ERROR   151
042526  000137  042564                  JMP     NN5
042532  023727  043102  043010  11$:    CMP     NNUD,#NNRP4     ;WAS IT THE RP4?
042540  001003                          BNE     12$
042542  104152                          ERROR   152
042544  000137  042564                  JMP     NN5
042550  104153                  12$:    ERROR   153
042552  000137  042564                  JMP     NN5

042556  062700  000004          2$:     ADD     #4,R0
042562  077140                          SOB     R1,1$

042564  005237  043110          NN5:    INC     NNGRPF          ;TESTED BOTH GROUPS?
042570  022737  000002  043110          CMP     #2,NNGRPF
042576  001410                          BEQ     NN6             ;BRANCH IF YES.
042600  012737  000044  043104          MOV     #S1M0,NNGRS     ;IF NOT GO BACK AND TEST GROUP ONE.
042606  012737  000030  043106          MOV     #S0M1,NNGRM
042614  000137  042444                  JMP     NN3

042620  000137  043476          NN6:    JMP     NNDONE

042624  104430                  NNRH4:  CALRH4          ;THIS IS THE CALL TO READ THE MASS BUS TESTER.
042626     071                          .BYTE   71
042627     000                  NNRH4U: .BYTE   0
042630  052525                          .WORD   52525
042632  000000                          .WORD   0
042634  140000                          .WORD   TESTR1
042636  000000                          .WORD   0
042640  001000                          .WORD   512.
042642  042654                          .WORD   2$

042644  005737  063500          1$:     TST     RH4ER1          ;ANY DEVICE ERRORS?
042650  100401                          BMI     2$              ;BRANCH IF YES.
042652  000207                          RTS     PC              ;IF NOT RETURN.

042654  013737  063502  001634  2$:     MOV     RH4ER2,$TMP1    ;REPORT DEVICE ERROR.
042662  013737  063504  001636          MOV     RH4ER3,$TMP2

042670  013737  063506  001640          MOV     RH4ER4,$TMP3
```

```
042676  005726                                 TST     (SP)+
042700  104156                                 ERROR   156
042702  105037  057462                         CLRB    RH4DFL
042706  105037  043114                         CLRB    RH4FT
042712  000137  042420                         JMP     NN1

042716  104426                        NNRS4:   CALRS4                          ;THIS IS A CALL TO DO AN RS4 READ.
042720     071                                 .BYTE   71
042721     000                        NNRS4U:  .BYTE   0
042722  000000                                 .WORD   0
042724  000000                                 .WORD   0
042726  140000                                 .WORD   TESTR1
042730  000000                                 .WORD   0
042732  001000                                 .WORD   512.
042734  042746                                 .WORD   2$

042736  005737  061004                1$:      TST     RS4ER1                  ;SEE IF THERE WERE DEVICE ERRORS.
042742  100401                                 BMI     2$                      ;BR IF YES.
042744  000207                                 RTS     PC

042746  013737  061006  001634        2$:      MOV     RS4ER2,$TMP1
042754  013737  061010  001636                 MOV     RS4ER3,$TMP2
042762  013737  061012  001640                 MOV     RS4ER4,$TMP3
042770  005726                                 TST     (SP)+
042772  104154                                 ERROR   154
042774  105037  057460                         CLRB    RS4DFL
043000  105037  043112                         CLRB    RS4FT
043004  000137  042420                         JMP     NN1

043010  104427                        NNRP4:   CALRP4                          ;THIS IS A CALL TO DO AN RP4 READ.
043012     071                                 .BYTE   71
043013     000                        NNRP4U:  .BYTE   0
043014  000000                                 .WORD   0
043016  000000                                 .WORD   0
043020  140000                                 .WORD   TESTR1
043022  000000                                 .WORD   0
043024  001000                                 .WORD   512.
043026  043040                                 .WORD   2$

043030  005737  060034                1$:      TST     RP4ER1                  ;WERE THERE ANY DEVICE ERRORS?
043034  100401                                 BMI     2$
043036  000207                                 RTS     PC

043040  013737  060036  001634        2$:      MOV     RP4ER2,$TMP1
043046  013737  060040  001636                 MOV     RP4ER3,$TMP2
043054  013737  060042  001640                 MOV     RP4ER4,$TMP3
043062  005726                                 TST     (SP)+
043064  104155                                 ERROR   155
043066  105037  057461                         CLRB    RP4DFL
043072  105037  043113                         CLRB    RP4FT
043076  000137  042420                         JMP     NN1

043102  000000                        NNUD:    .WORD   0

043104  000000                        NNGRS:   .WORD   0
043106  000000                        NNGRM:   .WORD   0
```

```
043110  000000                          NNGRPF: .WORD   0

                                        ;THIS ROUTINE IS CALLED TO MAKE THE ADDRESSES IN TESTR1
                                        ;HITS PRIOR TO CALLING FOR THE MB DEVICE TO DO TRANSFERS.
043112     000                          RS4FT:  .BYTE   0
043113     000                          RP4FT:  .BYTE   0
043114     000                          RH4FT:  .BYTE   0
043115     000                          RK5FT:  .BYTE   0
043116     000                          UBEFT:  .BYTE   0
        043120                                  .EVEN

043120  104412                          NNSTUP: SAVREG
043122  012700  043120                          MOV     #NNSTUP,R0      ;MAKE THIS CODE HITS IN THE
043126  012701  001000                          MOV     #512.,R1        ;GROUP NOT BEING TESTED.
043132  012702  142000                          MOV     #TESTR2,R2

043136  013737  043106  177746          1$:     MOV     NNGRM,@#CONTRL
043144  005720                                  TST     (R0)+
043146  013737  043104  177746                  MOV     NNGRS,@#CONTRL
043154  005722                                  TST     (R2)+
043156  077111                                  SOB     R1,1$

043160  013700  043104                  2$:     MOV     NNGRS,R0
043164  042700  000014                          BIC     #14,R0
043170  010037  177746                          MOV     R0,@#CONTRL
043174  012701  140000                          MOV     #TESTR1,R1
043200  012702  001000                          MOV     #512.,R2
043204  005721                          3$:     TST     (R1)+
043206  077202                                  SOB     R2,3$
043210  013700  043106                          MOV     NNGRM,R0
043214  042700  000014                          BIC     #14,R0
043220  010037  177746                          MOV     R0,@#CONTRL
043224  104413                                  RESREG
043226  000207                                  RTS     PC



                                        ;SEE WHAT DEVICE TO USE NEXT.
043230  000240                          NNDEV:  NOP
043232  000240                                  NOP
043234  005037  043102                          CLR     NNUD
043240  113700  043112                          MOVB    RS4FT,R0                ;IS THERE AN RS4 DRIVE.
043244  001430                                  BEQ     NND2                    ;BR IS NOT

043246  000240                          NND0:   NOP
043250  012701  000001                          MOV     #1,R1                   ;FIND OUT WHAT DRIVE  NUMBER IT IS.
043254  012737  042716  043102                  MOV     #NNRS4,NNUD
043262  005002                                  CLR     R2
043264  012703  000010                          MOV     #10,R3
043270  000240                          1$:     NOP
043272  030100                                  BIT     R1,R0
043274  001406                                  BEQ     2$
043276  140137  043112                          BICB    R1,RS4FT                ;FOUND IT.
043302  110237  042721                          MOVB    R2,NNRS4U
043306  000137  042424                          JMP     NN2
043312  005202                          2$:     INC     R2
043314  006301                                  ASL     R1
```

```
043316 077314                          SOB     R3,1$                   ;KEEP LOOKING.

043320 104000                          ERROR   0
043322 105037 043112                   CLRB    RS4FT

043326 000240                  NND2:   NOP
043330 113700 043113                   MOVB    RP4FT,R0                ;IS THERE AN RP04 DRIVE.
043334 001426                          BEQ     NND3                    ;BR IF NO
043336 012701 000001                   MOV     #1,R1
043342 012737 043010 043102            MOV     #NNRP4,NNUD
043350 005002                          CLR     R2
043352 012703 000010                   MOV     #10,R3
043356 030100                  1$:     BIT     R1,R0
043360 001406                          BEQ     2$
043362 140137 043112                   BICB    R1,RS4FT
043366 110237 043013                   MOVB    R2,NNRP4U
043372 000137 042424                   JMP     NN2
043376 005202                  2$:     INC     R2
043400 006301                          ASL     R1
043402 077313                          SOB     R3,1$
043404 104000                          ERROR   0
043406 105037 043113                   CLRB    RP4FT

043412 000240                  NND3:   NOP
043414 113700 043114                   MOVB    RH4FT,R0                ;IS THERE A MASS BUS TESTER.
043420 001426                          BEQ     NNDONE
043422 012701 000001                   MOV     #1,R1
043426 012737 042624 043102            MOV     #NNRH4,NNUD
043434 005002                          CLR     R2
043436 012703 000010                   MOV     #10,R3
043442 030100                  1$:     BIT     R1,R0
043444 001406                          BEQ     2$
043446 140137 043114                   BICB    R1,RH4FT
043452 110237 042627                   MOVB    R2,NNRH4U
043456 000137 042424                   JMP     NN2
043462 005202                  2$:     INC     R2
043464 006301                          ASL     R1
043466 077313                          SOB     R3,1$
043470 104000                          ERROR   0
043472 105037 043114                   CLRB    RH4FT
043476 104414                  NNDONE: RSET



043500 105737 001750                   TSTB    KB11CM                  ;11/74          (KB11CM)?
043504 001005                          BNE     1$                      ;BRANCH IF YES
043506 105737 001747                   TSTB    KB11EM                  ;KB11-EM?
043512 001002                          BNE     1$                      ;BR IF YES
043514 000137 050102                   JMP     KT                      ;GO TO KT IF NO
043520                         1$:                                     ;ENTER HERE IF KB11-E

                               ;;*************************************************************
                               ;*TEST 37        CHECK IVSS, VSIU BITS
                                       ;THIS TEST CHECKS THAT THE IVSS AND VSIU BITS OF THE CACHE
                                       ;CONTROL REGISTER CAN BE SET AND CLEARED.  VCIP IS ALSO
                                       ;CHECKED.
```

```
                                                 ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                         ;;**************************************************************
043520  000004                           TST37:  SCOPE
043522  005037  177746                           CLR     @#CONTRL
043526  005737  177746                           TST     @#CONTRL
043532  001404                                   BEQ     1$
043534  013737  177746  001562                   MOV     @#CONTRL,$REGO
043542  104055                                   ERROR   55      ;CCR COULD NOT BE CLEARED

043544  012737  040000  177746   1$:             MOV     #IVSS,@#CONTRL
043552  022737  040000  177746                   CMP     #IVSS,@#CONTRL
043560  001404                                   BEQ     2$
043562  013737  177746  001562                   MOV     @#CONTRL,$REGO
043570  104056                                   ERROR   56      ;IVSS COULD NOT BE SET

043572  042737  040000  177746   2$:             BIC     #IVSS,@#CONTRL
043600  001404                                   BEQ     3$
043602  013737  177746  001562                   MOV     @#CONTRL,$REGO
043610  104057                                   ERROR   57      ;IVSS COULD NOT BE CLEARED

043612  012737  020000  177746   3$:             MOV     #VSIU,@#CONTRL
043620  032737  020000  177746                   BIT     #VSIU,@#CONTRL
043626  001004                                   BNE     4$
043630  013737  177746  001562                   MOV     @#CONTRL,$REGO
043636  104060                                   ERROR   60      ;VSIU COULD NOT BE SET

043640  012700  000050           4$:             MOV     #50,R0  ;WAIT FOR VCIP TO CLEAR
043644  032737  010000  177746                   BIT     #VCIP,@#CONTRL
043652  001405                                   BEQ     5$
043654  077007                                   SOB     R0,4$
043656  013737  177746  001562                   MOV     @#CONTRL,$REGO
043664  104061                                   ERROR   61              ;VCIP DID NOT CLEAR WITHIN SOME
                                                                        ;SOME TIME AFTER VSIU WAS SET
043666  042737  020000  177746   5$:             BIC     #VSIU,@#CONTRL
043674  032737  020000  177746                   BIT     #VSIU,@#CONTRL
043702  001404                                   BEQ     6$
043704  013737  177746  001562                   MOV     @#CONTRL,$REGO
043712  104062                                   ERROR   62              ;VSIU COULD NOT BE CLEARED
043714  032737  010000  177746   6$:             BIT     #VCIP,@#CONTRL
043722  001374                                   BNE     6$
                                         ;;**************************************************************
                                         ;*TEST 40        CHECK VSIU BIT, WITH IVSS ALREADY SET
                                                 ;THIS TEST CHECKS THAT THE 'VALID STORE IN USE'' (VISU)
                                                 ;BIT CAN BE SET AND CLEARED WHEN THE IVSS IS
                                                 ;ALREADY SET.
                                                 ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                         ;;**************************************************************
043724  000004                           TST40:  SCOPE

043726  012737  040000  177746                   MOV     #IVSS,@#CONTRL
043734  032737  020000  177746                   BIT     #VSIU,@#CONTRL
043742  001404                                   BEQ     1$
043744  013737  177746  001562                   MOV     @#CONTRL,$REGO
043752  104062                                   ERROR   62              ;VALID STORE IN USE, BIT 13,
                                                                        ;COULD NOT BE CLEARED IN CCR
043754  032737  010000  177746   1$:             BIT     #VCIP,@#CONTRL
```

```
043762 001374                           BNE     1$
043764 052737 020000 177746             BIS     #VSIU,@#CONTRL
043772 032737 020000 177746             BIT     #VSIU,@#CONTRL
044000 001004                           BNE     2$
044002 013737 177746 001562             MOV     @#CONTRL,$REG0
044010 104060                           ERROR   60              ;VSIU (BIT 13) COULD NOT BE SET
                                                                ;IN CCR (IVSS WAS ALREADY SET).
044012 042737 020000 177746     2$:     BIC     #VSIU,@#CONTRL
044020 032737 020000 177746             BIT     #VSIU,@#CONTRL
044026 001404                           BEQ     TST41           ;;EXIT
044030 013737 177746 001562             MOV     @#CONTRL,$REG0
044036 104062                           ERROR   62              ;VSIU COULD NOT BE CLEARED IN CCR
                                                                ;IVSS WAS ALREADY SET.

                                ;;*************************************************************
                                ;*TEST 41       CHECK VCIP SETS WHEN CF IS SET
                                        ;THIS TEST CHECKS THAT THE VCIP SETS WHEN 'ACHE-FLUSH IS
                                        ;DONE AND IT CLEARS OUT WITHIN A CERTAIN TIME ArTER
                                        ;THE FLUSH OF VALID STORE IS OVER
                                         ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                ;;*************************************************************
044040 000004                   TST41:  SCOPE
044042 012737 000400 177746             MOV     #FCAC, @# CONTRL;FLUSH CACHE
044050 000240                           NOP
044052 012700 000062                    MOV     #50.,R0
044056 032737 010000 177746             BIT     #VCIP, @# CONTRL
044064 001004                           BNE     1$
044066 013737 177746 001562             MOV     @#CONTRL,$REG0
044074 104063                           ERROR   63      ;VCIP DID NOT SET WHEN CACHE
                                                        ;FLUSH WAS ISSUED
044076 032737 010000 177746     1$:     BIT     #VCIP, @#CONTRL ;WAIT FOR VCIP TO CLEAR
044104 001405                           BEQ     2$
044106 077005                           SOB     R0,1$
044110 013737 177746 001562             MOV     @#CONTRL,$REG0
044116 104061                           ERROR   61              ;VCIP DID NOT CLEAR WITHIN A
                                                                ;CERTAIN TIME AFTER CACHE FLUSH
                                                                ;WAS DONE
044120                          2$:

                                ;;*************************************************************
                                ;*TEST 42       CHECK CACHE FLUSH & VALID STORE SWITCHING
                                        ;THIS TEST CHECKS THAT WHEN A CACHE FLUSH IS DONE
                                        ;BY SETTING CF IN CCR, THE VALID STORE IN USE
                                        ;(VSIU) SURTCHES. VALID STORE SWITCHING FROM STORE-A
                                        ;TO STORE-B AND VICE-VERSA IS CHECKED
                                         ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                ;;*************************************************************
044120 000004                   TST42:  SCOPE
044122 005037 177746                    CLR     @#CONTRL
044126 032737 010000 177746     1$:     BIT     #VCIP, @#CONTRL
044134 001374                           BNE     1$
044136 012737 000400 177746             MOV     #FCAC, @#CONTRL ;FLUSH CACHE
044144 032737 020000 177746             BIT     #VSIU, @#CONTRL
044152 001004                           BNE     2$
044154 013737 177746 001562             MOV     @#CONTRL,$REG0
044162 104064                           ERROR   64      ;VSIU DID NOT SWITCH FROM 0 TO 1
```

```
8133                                                        ;WHEN CACHE FLUSH WAS SET
8134  044164  032737  010000  177746  2$:      BIT     #VCIP, @#CONTRL
8135  044172  001374                           BNE     2$
8136  044174  012737  000400  177746           MOV     #FCAC, @#CONTRL
8137  044202  032737  020000  177746           BIT     #VSIU, @#CONTRL
8138  044210  001404                           BEQ     3$
8139  044212  013737  177746  001562           MOV     @#CONTRL,$REGO
8140  044220  104064                           ERROR   64       ;VSIU DID NOT SWITCH FROM 1 TO 0 WHEN
8141                                                        ;FLUSH-CACHE WAS SET IN CLR
8142  044222  032737  010000  177746  3$:      BIT     #VCIP, @#CONTRL
8143  044230  001374                           BNE     3$
8144
8145
8146                                      ;;*************************************************************
8147                                      ;;*TEST 43        CHECK IVSS INHIBITS SWITCHING OF VALID STORE IN USE
8148                                               ;THIS TEST CHECKS THAT WHEN ''INHIBIT VALID STORE SWITCHING''
8149                                               ;(IVSS) IS SET AND FLUSH-CACHE BIT IS SET, THE
8150                                               ;VALID STORE IN USE DOES NOT SWITCH
8151                                               ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
8152                                      ;;*************************************************************
8153  044232  000004                      TST43:   SCOPE
8154  044234  005037  177746                       CLR     @#CONTRL
8155  044240  032737  010000  177746  1$:      BIT     #VCIP,@#CONTRL
8156  044246  001374                           BNE     1$
8157  044250  012737  040000  177746           MOV     #IVSS, @#CONTRL ;SET IVSS
8158  044256  052737  000400  177746           BIS     #FCAC, @#CONTRL ;FLUSH CACHE
8159  044264  032737  020000  177746           BIT     #VSIU, @#CONTRL
8160  044272  001404                           BEQ     2$              ;CHECK VSIU DID NOT SWITH
8161  044274  013737  177746  001562           MOV     @#CONTRL,$REGO
8162  044302  104065                           ERROR   65               ;VSIU SWITCHED, WHEN IVSS
8163                                                            ;WAS SET AND CACHE FLUSH
8164                                                            ;WAS DONE, IT SHOULD NOT SWITCH
8165  044304  032737  010000  177746  2$:      BIT     #VCIP, @#CONTRL
8166  044312  001374                           BNE     2$
8167  044314  052737  020000  177746           BIS     #VSIU, @#CONTRL
8168  044322  032737  010000  177746  3$:      BIT     #VCIP, @#CONTRL
8169  044330  001374                           BNE     3$
8170  044332  052737  000400  177746           BIS     #FCAC, @#CONTRL
8171  044340  032737  020000  177746           BIT     #VSIU, @#CONTRL ;CHECK VSIU DID NOT SWITCH
8172  044346  001004                           BNE     4$
8173  044350  013737  177746  001562           MOV     @#CONTRL,$REGO
8174  044356  104065                           ERROR   65               ;VSIU SWITCHED, WHEN IVSS
8175                                                            ;WAS SET AND CACHE FLUSH WAS
8176                                                            ;DONE; IT SHOULD NOT SWITCH
8177  044360  032737  010000  177746  4$:      BIT     #VCIP, @#CONTRL
8178  044366  001374                           BNE     4$
8179
8180
8181                                      ;;*************************************************************
8182                                      ;;*TEST 44        CHECK VALID STORES (A & B) FOR GROUP 0
8183                                               ;THIS TEST CHECKS THE TWO VALID STORES (A&B) FOR GROUP 0
8184                                               ;OF THE CACHE. WHEN A CACHE-FLUSH IS ISSUED, THE CACHE
8185                                               ;SHOULD BE INVALIDATED BY SWITCHING THE VALID STORE
8186                                               ;IN USE
8187                                               ;THE TEST-CODE IS MADE HIT IN GROUP 1 (WHICH IS NOT
8188                                               ;BEING TESTED). THE TEST DATA IS MADE HIT IN GROUP 0.
```

```
                                                   ;FLUSH-CACHE BIT IS SET IN THE CCR. IT IS CHECKED THAT
                                                   ;THE TEST-DATA WHICH WAS HIT (MADE PREVIOUSLY) IN
                                                   ;GROUP 0 IS NO MORE A HIT. EACH LOCATION OF THE
                                                   ;TEST-DATA BLOCK IS REFERENCED AND CHECKED IF
                                                   ;IT WAS A MISS. OTHERWISE AN ERROR IS REPORTED. AS A
                                                   ;RESULT OF THE CACHE FLUSH THE VALID STORE SHOULD
                                                   ;HAVE SWITCHED FROM 0 TO 1. THEN THE VALID STORE
                                                   ;IS FORCED TO BE 0 AND THE TEST-DATA IS REFERENCED
                                                   ;AGAIN. IT IS CHECKED IF IT WAS A MISS.
                                                    ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR FM
                                           ;;*****************************************************************
044370  000004                             TST44:  SCOPE
044372  005005                             VSG0:   CLR     R5
044374  010537  177746                     VSG0A:  MOV     R5, @#CONTRL
044400  032737  010000  177746                     BIT     #VCIP, @#CONTRL
044406  001374                                     BNE     .-6
044410  012702  000034                             MOV     #S0M0M1,R2
044414  012703  000054                             MOV     #S1M0M1,R3
044420  050502                                     BIS     R5,R2
044422  050503                                     BIS     R5,R3
044424  012700  044372                             MOV     #VSG0, R0       ;MAKE TEST-CODE HIT IN
044430  012701  001000                             MOV     #1000, R1       ;GROUP 1
044434  010237  177746             1$:     MOV     R2, @#CONTRL;FORCE REPLACE GROUP 0
044440  005762  002000                             TST     2000(R2)
044444  010337  177746                             MOV     R3, @#CONTRL   ;FORCE REPLACE GROUP 1
044450  005720                                     TST     (R0)+
044452  077110                                     SOB     R1, 1$
044454  012700  116310                             MOV     #TSTDAT, R0     ;MAKE TEST-DATA HIT IN
044460  012701  001000                             MOV     #1000, R1       ;GROUP 0
044464  010337  177746                             MOV     R3, @#CONTRL
044470  042737  000014  177746                     BIC     #M0M1, @#CONTRL   ;FORCE REPLACE GROUP 0
044476  005720                     2$:             TST     (R0)+
044500  077102                                     SOB     R1, 2$
044502  042737  000020  177746                     BIC     #S0, @#CONTRL
044510  052737  000040  177746                     BIS     #S1, @#CONTRL   ;FORCE REPLACE GROUP 1
044516  052737  000400  177746                     BIS     #FCAC,@#CONTRL  ;FLUSH CACHE
044524  013704  177746                             MOV     @#CONTRL, R4
044530  074504                                     XOR     R5, R4          ;CHECK IF VSIU COMPLEMENTED
044532  032704  020000                             BIT     #VSIU, R4
044536  001004                                     BNE     3$
044540  013737  177746  001562                     MOV     @#CONTRL,$REG0
044546  104064                                     ERROR   64              ;VSIU DID NOT SWITCH WHEN
                                                                           ;CACHE-FLUSH WAS DONE
044550  052737  000014  177746     3$:             BIS     #M0M1, @#CONTRL ;MAKE TEST-CODE HIT IN
044556  012700  044372                             MOV     #VSG0, R0       ;GROUP 1
044562  012701  001000                             MOV     #1000,R1
044566  005720                     4$:             TST     (R0)+
044570  077102                                     SOB     R1, 4$
044572  042737  000014  177746                     BIC     #M0M1, @#CONTRL
044600  012700  116310                             MOV     #TSTDAT, R0      ;REFERENCE TEST-DATA AND CHECK
044604  012701  000400                             MOV     #400, R1        ;THAT IT IS A MISS. NOTE
044610  005710                     5$:             TST     (R0)            ;SETTING CACHE-FLUSH SHOULD
044612  032737  000010  177752                     BIT     #10, @#HITMIS    ;HAVE INVALIDATED GROUP 0
044620  001410                                     BEQ     6$
044622  013737  177746  001562                     MOV     @#CONTRL,$REG0
044630  005037  001564                             CLR     $REG1            ;GROUP NO.
```

```
044634  010037  001566                    MOV     R0,$REG2        ;TEST DATA ADDRESS
044640  104066                            ERROR   66              ;TEST-DATA WAS NOT A MISS.
                                                                  ;TEST DATA WAS MADE A HIT
                                                                  ;IN GROUP 0 AND THEN CACHE-
                                                                  ;FLUSH WAS DONE. CACHE-FLUSH
                                                                  ;SHOULD HAVE INVALIDATED GROUP
                                                                  ;0'S CACHED DATA, HENEE, THE
                                                                  ;TEST DATA REFERENCE SHOULD
                                                                  ;HAVE BEEN A MISS.
                                                                  ;PROBLE FAULURE
044642  062700  000004            6$:     ADD     #4, R0          ;VALID STORE IS NOT BEING SWITCHED
044646  077120                            SOB     R1, 5$          ;TO THE OTHER WHEN CACHE-FLUSH IS
                                                                  ;SET IN THE CCR
044650  032737  010000  177746    7$:     BIT     #VCIP, @#CONTRL
044656  001374                            BNE     7$
044660  012700  020000                    MOV     #VSIU, R0        ;COMPLEMENT VSIU
044664  074037  177746                    XOR     R0, @#CONTRL
044670  032737  010000  177746    8$:     BIT     #VCIP, @#CONTRL
044676  001374                            BNE     8$
044700  052737  000014  177746            BIS     #MOM1, @#CONTRL ;MAKE TEST-CODE HIT IN
044706  012700  044372                    MOV     #VSG0, R0        ;GROUP 1
044712  012701  001000                    MOV     #1000, R1
044716  005720                     9$:    TST     (R0)+
044720  077102                            SOB     R1, 9$
044722  042737  000014  177746            BIC     #MOM1, @#CONTRL
                                                                   ;THE ORIGINAL VALID STORE (WHICH
                                                                   ;WAS INVALIDATED BY CACHE FLUSH)
                                                                   ;IS IN USE AGAIN.
044730  012700  116310                    MOV     #TSTDAT, R0
044734  012701  000400                    MOV     #400, R1         ;REFERENCE THE TEST-DATA AND
                                                                   ;CHECK IT IS A MISS
044740  005710                    10$:    TST     (R0)
044742  032737  000010  177752            BIT     #10, @#HITMIS
044750  001410                            BEQ     11$
044752  013737  177746  001562            MOV     @#CONTRL,$REG0
044760  005037  001564                    CLR     $REG1            ;GROUP NO.
044764  010037  001566                    MOV     R0,$REG2         ;TEST DATA ADDRESS
044770  104067                            ERROR   67          ;TEST-DATA REFERENCE WAS NOT A MISS (IN
                                                              ;GROUP 0, ORIGINAL VALID STORE). CACHE-FLUSH
                                                              ;DONE EARLIER ON THE ORIGINAL VALID STORE
                                                              ;SHOULD HAVE RESULTED IN INVALIDATING
                                                              ;THE VALID STORE, THUS RESULTING IN
                                                              ;CACHE-MISS ON TEST DATA REFERENCE.
                                                              ;PROBALE FAILURE: VALID STORE IN USE IS NOT
                                                              ;BEING INVALIDATD WHEN CACHE-FLUSH IS
                                                              ;SET
044772  062700  000004            11$:    ADD     #4, R0
044776  077120                            SOB     R1, 10$
045000  012701  020000                    MOV     #VSIU,R1
045004  074105                            XOR     R1,R5       ;TESTED VALID STORE B (1)?
045006  001402                            BEQ     TST45            ;;EXIT
045010  000137  044374                    JMP     VSG0A



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

```
                                                     ;THIS TEST CHECKS THAT HIT CAN BE OBTAINED FROM BOTH GROUPS
                                                     ;(0&1) OF THE CACHE, FROM EACH OF THE TWO VALID
                                                     ;STORES (A&B) PER GROUP. THUS ALL 4 VALID STORES GET
                                                     ;CHECKED.
                                                     ;TEST-DATA (UNIQUE) IS MADE A HIT IN GROUP 0 USING
                                                     ;THE FIRST VALID STORE A. TEST-CODE IS MADE A HIT IN THE
                                                     ;GROUP NOT BEING TESTED. TEST-DATA IS READ BACK AND
                                                     ;CHECKED FOR CORRECTNESS. IT IS ALSO CHECKED IF THE
                                                     ;TEST-DATA REFERENCE WAS A HIT. THE TESTING IS
                                                     ;REPEATED FOR VALID STORE B.
                                                     ;THE ENTIRE TEST (ABOVE) IS REPEATED FOR GROUP 1.
                                                     ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                             ;;*********************************************************************
 045014  000004                              TST45:  SCOPE
 045016  005002                              G1G0V:  CLR     R2      ;VSIU HIT MASK
 045020  005005                                      CLR     R5      ;INITIALIZE COUNT DATA PATTERN TO BE USED
 045022  012700  000034                      G1G0VA: MOV     #S0M0M1, R0
 045026  012701  000054                              MOV     #S1M0M1, R1
 045032  010237  177746                      G1G0VB: MOV     R2, @#CONTRL
 045036  032737  010000  177746              1$:     BIT     #VCIP,@#CONTRL
 045044  001374                                      BNE     1$
 045046  050200                                      BIS     R2,R0
 045050  050201                                      BIS     R2,R1
 045052  012703  045016                              MOV     #G1G0V, R3      ;MAKE TEST-CODE HIT IN THE
 045056  012704  001000                              MOV     #1000, R4       ;GROUP NOT BEINNG TESTED
 045062  010037  177746                      2$:     MOV     R0, @#CONTRL
 045066  005763  002000                              TST     2000 (R3)
 045072  010137  177746                              MOV     R1, @#CONTRL
 045076  005723                                      TST     (R3)+
 045100  077410                                      SOB     R4, 2$
 045102  042700  000014                              BIC     #M0M1, R0       ;WRITE COUNT PATTERN AND MAKE
 045106  042701  000014                              BIC     #M0M1, R1       ;IT A HIT IN THE GROUP BEING
 045112  012703  116310                              MOV     #TSTDAT, R3     ;TESTED.
 045116  012704  001000                              MOV     #1000, R4       ;BIT 15 OF THE COUNT PATTERN INDICATES
                                                                             ;WHICH GROUP;BIT15=0, GROUP 0, ELSE 1
 045122  010037  177746                              MOV     R0,@#CONTRL     ;BIT 14 OF THE COUNT PATTERN INDICATES
 045126  010513                              3$:     MOV     R5, (R3)       ;WHICH VALID STORE, A (0) OR B (1)
 045130  005723                                      TST     (R3)+           ;MAKE IT A HIT
 045132  005205                                      INC     R5
 045134  077404                                      SOB     R4, 3$
 045136  010137  177746                              MOV     R1, @#CONTRL
 045142  012703  116310                              MOV     #TSTDAT, R3
 045146  012704  001000                              MOV     #1000,  R4
 045152  042705  001777                              BIC     #1777, R5        ;INITIALIZE PATTERN TO BE CHECKED
 045156  011337  045332                      4$:     MOV     (R3), TMP       ;READ THE TEST-DATA AND
 045162  032737  000020  177752                      BIT     #20, @#HITMIS   ;CHECK IF THE REFERENCE WAS
 045170  001016                                      BNE     5$             ;A HIT
 045172  013737  177746  001562                      MOV     @#CONTRL,$REG0
 045200  005037  001564                              CLR     $REG1           ;GROUP NO.
 045204  032705  100000                              BIT     #BIT15,R5       ;WHICH GROUP?
 045210  001403                                      BEQ     8$
 045212  012737  000001  001564                      MOV     #1,$REG1
 045220  010337  001566                      8$:     MOV     R3,$REG2         ;TEST DATA ADDRESS
 045224  104070                                      ERROR   70             ;TEST-DATA REFERENCE WAS NOT A
                                                                            ;HIT, FROM THE GROUP AND
                                                                            ;VALID STORE BEING TESTED
```

```
045226  023705  045332          5$:     CMP     TMP, R5         ;DATA CORRECT?
045232  001410                          BEQ     6$
045234  010537  001562                  MOV     R5,$REG0         ;EXPCTD DATA
045240  013737  045332  001564          MOV     TMP,$REG1        ;DATA RECVD
045246  010337  001566                  MOV     R3,$REG2
045252  104071                          ERROR   71              ;READ INCORRECT DATA ON REFEREN
                                                                ;-CING A CACHED LOCATION.
045254  062703  000002          6$:     ADD     #2, R3
045260  005205                          INC     R5
045262  077443                          SOB     R4, 4$
045264  012704  020000                  MOV     #VSIU,R4
045270  074402                          XOR     R4, R2        ;DONE VALID STORE B (1)?
045272  001405                          BEQ     7$              ;YES
045274  052705  040000                  BIS     #BIT14, R5      ;INDICATE VS-B IN DATA-PATTERN
045300  042705  001777                  BIC     #1777,R5
045304  000646                          BR      G1G0VA          ;CHECK GROUP, VS-B
045306  032705  100000          7$:     BIT     #BIT15, R5       ;DONE CHECKING GROUP 1?
045312  001010                          BNE     TST46            ;;EXIT
045314  012700  000054                  MOV     #S1M0M1, R0
045320  012701  000034                  MOV     #S0M0M1, R1
045324  012705  100000                  MOV     #BIT15, R5 ;INDICATE GROUP 1
045330  000640                          BR      G1G0VB
045332  000000                  TMP:    .WORD   0


                                ;;*****************************************************************
                                ;*TEST 46        CHECK VALID STORES (A &B ) FOR GROUP 1
                                         ;THIS TEST CHECKS RTHE TWO VALID STORES (A&B) FOR GROUP 1
                                        ;OF THE CACHE. WHEN A CACHE-FLUSH IS ISSUED, THE CACHE
                                        ;SHOULD BE INVALIDATED BY SWITCHING THE VALID STORE
                                        ;IN USE.
                                        ;THE TEST-CODE IS MADE HIT IN GROUP 1 (WHICH IS NOT
                                        ;BEING TESTED). THE TEST DATA IS MADE HIT IN GROUP 0.
                                        ;FLUSH-CACHE HIT IS SET IN THE CCR. IT IS CHECKED THAT
                                        ;THE TEST-DATA WHICH WAS HIT (MADE PRECIOUSLY) IN
                                        ;GROUP 0 IS NO MORE A HIT, EACH LOCATION OF THE
                                        ;TEST-DATA BLOCK IS REFERENCED AND CHECKED IF
                                        ;IT WAS A MISS. OTHERWISE AN ERROR IS REPORTED. AS A
                                        ;RESULT OF THE CACHE FLUSH THE VALID STORE SHOULD
                                        ;HAVE SWITCHED FROM 0 TO 1. THEN THE VALID STORE
                                        ;IS FORCED TO BE 0 AND THE TEST-DATA IS REFERENCED
                                        ;AGAIN. IT IS CHECKED IF IT WAS A MISS.
                                        ;THE WHOLE TEST IS REPEATED USING VALID-STORE
                                        ;B (1).
                                         ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                ;;*****************************************************************
045334  000004                  TST46:  SCOPE
045336  005005                  VSG1:   CLR     R5               ;R5, HIT MASK FOR VSIU
045340  010537  177746          VSG1A:  MOV     R5, @#CONTRL
045344  032737  010000  177746          BIT     #VCIP,@#CONTRL
045352  001374                          BNE     .-6
045354  012702  000034                  MOV     #S0M0M1,R2
045360  012703  000054                  MOV     #S1M0M1,R3
045364  050502                          BIS     R5,R2
045366  050503                          BIS     R5,R3
045370  012700  045336                  MOV     #VSG1, R0        ;MAKE TEST-CODE HIT IN
```

```
045374  012701  001000                   MOV     #1000, R1          ;GROUP 0
045400  010337  177746          1$:      MOV     R3, @#CONTRL;FORCE REPLACE GROUP 1
045404  005760  002000                   TST     2000(R0)
045410  010237  177746                   MOV     R2, @#CONTRL   ;FORCE REPLACE GROUP 0
045414  005720                           TST     (R0)+
045416  077110                           SOB     R1, 1$
045420  012700  116310                   MOV     #TSTDAT, R0     ;MAKE TEST-DATA HIT IN
045424  012701  001000                   MOV     #1000, R1       ;GROUP 1
045430  010337  177746                   MOV     R3, @#CONTRL   ;FORCE REPLACE GROUP 1
045434  042737  000014  177746           BIC     #MOM1,@#CONTRL
045442  005720                  2$:      TST     (R0)+
045444  077102                           SOB     R1, 2$
045446  042737  000040  177746           BIC     #S1, @#CONTRL
045454  052737  000020  177746           BIS     #S0, @#CONTRL   ;FORCE REPLACE GROUP 0
045462  052737  000400  177746           BIS     #FCAC, @#CONTRL ;FLUSH CACHE
045470  013704  177746                   MOV     @#CONTRL, R4
045474  074504                           XOR     R5, R4          ;CHECK IF VSIU COMPLEMENTED
045476  032704  020000                   BIT     #VSIU, R4
045502  001004                           BNE     3$
045504  013737  177746  001562           MOV     @#CONTRL,$REG0
045512  104064                           ERROR   64             ;VSIU DID NOT SWITCH WHEN
                                                                ;CACHE-FLUSH WAS DONE
045514  052737  000014  177746  3$:      BIS     #MOM1,  @#CONTRL;MAKE TEST-CODE HIT IN
045522  012700  045336                   MOV     #VSG1,R0                  ;GROUP0
045526  012701  001000                   MOV     #1000,R1
045532  005720                  4$:      TST     (R0)+
045534  077102                           SOB     R1, 4$
045536  042737  000014  177746           BIC     #MOM1,@#CONTRL
045544  012700  116310                   MOV     #TSTDAT, R0     ;REFERENCE TEST-DATA AND CHECK
045550  012701  000400                   MOV     #400,R1        ;THAT IT IS A MISS. NOTE
045554  005710                  5$:      TST     (R0)           ;SETTING CACHE-FLUSH SHOULD
045556  032737  000010  177752           BIT     #10, @#HITMIS   ;HAVE INVALIDATED GROUP
045564  001411                           BEQ     6$
045566  013737  177746  001562           MOV     @#CONTRL,$REG0
045574  012737  000001  001564           MOV     #1,$REG1        ;GROUP NO.
045602  010037  001566                   MOV     R0,$REG2        ;TEST DATA ADDRESS
045606  104066                           ERROR   66             ;TEST-DATA WAS NOT A MISS.
                                                                ;TEST DATA WAS MADE A HIT
                                                                ;IN GROUP 1 AND THEN CACHE-
                                                                ;FLUSH WAS DONE. CACHE-FLUSH
                                                                ;SHOULD HAVE INVALIDATED GROUP
                                                                ;1'S CACHED DATA. HENEE, THE
                                                                ;TEST DATA REFERENCE SHOULD
                                                                ;HAVE BEEN A MISS.
                                                                ;PROBABLE FAILURE:
045610  062700  000004          6$:      ADD     #4, R0         ;VALID STORE IS NOT REING SWITCHED
045614  077121                           SOB     R1, 5$         ;TO THE OTHER WHEN CACHE-FLUSH IS
                                                                ;SET IN THE CCR
045616  032737  010000  177746  7$:      BIT     #VCIP, @#CONTRL
045624  001374                           BNE     7$
045626  012700  020000                   MOV     #VSIU, R0       ;COMPLEMENT VSIU
045632  074037  177746                   XOR     R0, @#CONTRL
045636  032737  010000  177746  8$:      BIT     #VCIP, @#CONTRL
045644  001374                           BNE     8$
045646  052737  000014  177746           BIS     #MOM1, @#CONTRL ;MAKE TEST-CODE HIT IN
045654  012700  045336                   MOV     #VSG1, R0       ;GROUP 0
```

```
045660 012701  001000                 MOV     #1000, R1
045664 005720                 9$:     TST     (R0)+
045666 077102                         SOB     R1, 9$
045670 042737  000014  177746         BIC     #MOM1, @#CONTRL
                                                              ;THE ORIGINAL VALID STORE (WHICH
                                                              ;WAS INVALIDATED BY CACHE FLUSH)
                                                              ;IS IN USE AGAIN.
045676 012700  116310                 MOV     #TSTDAT, R0
045702 012701  000400                 MOV     #400, R1        ;REFERENCE THE TEST-DATA AND
                                                              ;CHECK IT IS A MISS
045706 005710                 10$:    TST     (R0)
045710 032737  000010  177752         BIT     #10, @#HITMIS
045716 001411                         BEQ     11$
045720 013737  177746  001562         MOV     @#CONTRL,$REG0
045726 012737  000001  001564         MOV     #1,$REG1          ;GROUP NO.
045734 010037  001566                 MOV     R0,$REG2          ;TEST DATA ADDRESS
045740 104067                         ERROR   67          ;TEST-DATA REFERENCE WAS NOT A MISS (IN
                                                         ;GROUP 1, ORIGINAL VALID STORE). CACHE-FLUSH
                                                         ;DONE EARLIER ON THE ORIGINAL VALID STORE
                                                         ;SHOULD HAVE RESULTED IN INVALIDATING
                                                         ;THE VALID STORE, THUS RESULTING IN
                                                         ;CACHE-MISS ON TEST DATA REFERENCE.
                                                         ;PROBABLE FAILURE: VALID STORE IN USE IS NOT
                                                         ;BEING INVALIDATED WHEN CACHE-FLUSH IS
                                                         ;SET
045742 062700  000004         11$:    ADD     #4, R0
045746 077121                         SOB     R1, 10$
045750 012701  020000                 MOV     #VSIU,R1
045754 074105                         XOR     R1, R5 ;TESTED VALID STORE B (1)?
045756 001402                         BEQ     TST47             ;;EXIT
045760 000137  045340                 JMP     VSG1A


                              ;;*****************************************************************
                              ;*TEST 47        CHECK CACHE TURNS OFF WHEN FLUSH IS DONE WITH IVSS SET
                                      ;THIS TEST CHECKS THAT IF CACHE-FLUSH IS DONE (SETTING CF), WHEN IVSS
                                      ;IS SET, THE VALID STORES ARE NOT SWITCHED AND THE CACHE IS TURNED
                                      ;OFF (AND A SLOW FLUSH IS PERFORMED). THUS, ANY REFERENCE TO
                                      ;A PREVIOUSLY CACHED DATA SHOULD RESULT IN CACHE MISS.
                                      ;TEST-DATA IS MADE HIT IN GROUP 0 (BEING TESTED). TEST CODE IS
                                      ;MADE HIT IN GROUP 1.IVSS IS SET AND A FLUSH IS DONE. PREVIOUSLY
                                      ;CACHED TEST-DATA IS REFERENCED TO CHECK IT IS A MISS.
                                      ;THE TEST IS REPEATED FOR BOTH GROUPS AND VALID STORES.
                                      ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                              ;;*****************************************************************
045764 000004                 TST47:  SCOPE

045766 005002                 IVFC:   CLR     R2                ;BIT MASK FOR VSIU
045770 012700  000034                 MOV     #S0MOM1,R0
045774 012701  000054                 MOV     #S1MOM1,R1
046000 050200                 IVFCA:  BIS     R2,R0
046002 050201                         BIS     R2,R1
046004 010237  177746                 MOV     R2,@#CONTRL
046010 032737  010000  177746 1$:     BIT     #VCIP,@#CONTRL
046016 001374                         BNE     1$
```

```
046020  012703  045766                    MOV     #IVFC,R3                ;MAKE TEST CODE BIT IN GROUP
046024  012704  001000                    MOV     #1000,R4        ;NOT BEING TESTED
046030  010037  177746            2$:     MOV     R0,@#CONTRL
046034  005763  002000                    TST     2000(R3)
046040  010137  177746                    MOV     R1,@#CONTRL
046044  005723                            TST     (R3)+
046046  077410                            SOB     R4,2$

046050  042700  000014                    BIC     #MOM1,R0
046054  042701  000014                    BIC     #MOM1,R1

046060  012703  116310                    MOV     #TSTDAT,R3      ;MAKE TEST-DATA HIT IN GROUP
046064  012704  001000                    MOV     #1000,R4        ;BEING TESTED

046070  010037  177746                    MOV     R0,@#CONTRL
046074  005723                    3$:     TST     (R3)+
046076  077402                            SOB     R4, 3$

046100  010137  177746                    MOV     R1,@#CONTRL      ;FORCE REPLACE GRCUP (NOT BEING TESTED)
046104  052737  040000  177746            BIS     #IVSS,@#CONTRL  ;SET IVSS

046112  012705  000004                    MOV     #4,R5           ;BIT MASK FOR HIT/MISS REGISTER
046116  012704  000400                    MOV     #400,R4
046122  012703  120310                    MOV     #TSTDAT+2000,R3
046126  052737  000400  177746            BIS     #FCAC,@#CONTRL  ;FLUSH CACHE
046134  005743                    4$:     TST     -(R3)           ;REFERENCE PREVIOUSLY CACHED
046136  030537  177752                    BIT     R5,@#HITMIS     ;TEST DATA& CHECK IT IS A MISS
046142  001004                            BNE     6$
046144  162703  000002            5$:     SUB     #2,R3
046150  077407                            SOB     R4,4$
046152  000417                            BR      7$              ;DONE

046154  013737  177746  001562    6$:     MOV     @#CONTRL,$REG0
046162  005037  001564                    CLR     $REG1           ;GROUP NO.
046166  032700  000040                    BIT     #S1,R0          ;WHICH GROUP?
046172  001403                            BEQ     12$
046174  012737  000001  001564            MOV     #1,$REG1        ;GROUP NO
046202  010337  001566            12$:    MOV     R3,$REG2        ;TEST DATA ADDRESS
046206  104072                            ERROR   72              ;TEST DATA REFERENCE DID NOT
                                                                  ;REGISTER A MISS. TEST-DATA WAS
046210  000755                            BR      5$              ;MADE BIT IN A GROUP. CACHE-FLUSH
                                                                  ;WAS DONE, WITH IVSS SET. REFERENCE
                                                                  ;TO THE PREVIOUSLY CACHED DATA
                                                                  ;SHOULD HAVE BEEN A MISS.
                                                                  ;PROBABLE FAILURE:  CACHE DOES NOT
                                                                  ;TURN OFF WHEN IVSS IS SET AND
                                                                  ;FLUSH IS PERFORMED
                                          ;CHECK THAT THE CACHE HAS TURNED ON AGAIN,CHECK
                                          ;THAT HITS CAN BE OBTAINED

046212  012703  045766            7$:     MOV     #IVFC,R3        ;MAKE THE TEST-CODE HIT IN GROUP NOT
046216  012704  001000                    MOV     #1000,R4        ;BEING TESTED
046222  052700  000014                    BIS     #MOM1,R0
046226  052701  000014                    BIS     #MOM1,R1
046232  010037  177746            8$:     MOV     R0,@#CONTRL
046236  005763  002000                    TST     2000(R3)
```

```

046242  010137  177746                    MOV     R1,@#CONTRL
046246  005723                            TST     (R3)+
046250  077410                            SOB     R4,8$

046252  042700  000014                    BIC     #MOM1,R0
046256  042701  000014                    BIC     #MOM1,R1

046262  012703  116310                    MOV     #TSTDAT,R3      ;MAKE TEST-DATA HIT IN GROUP
046266  012704  001000                    MOV     #1000,R4        ;BEING TESTED

046272  010037  177746                    MOV     R0,@#CONTRL
046276  005723                    9$:     TST     (R3)+
046300  077402                            SOB     R4,9$

                                                                  ;FORCE REPLACE GROUP NOT BEING
046302  010137  177746                    MOV     R1,@#CONTRL     ;TESTED
046306  012703  116310                    MOV     #TSTDAT,R3      ;REFERENCE TEST-DATA (IN THE
046312  012704  001000                    MOV     #1000,R4        ;GROUP BEING CHECKED) AND
046316  005713                    10$:    TST     (R3)            ;MAKE SURE IT IS A HIT
046320  032737  000010  177752            BIT     #10,@#HITMIS    ;HIT?
046326  001016                            BNE     11$             ;YES
046330  013737  177746  001562            MOV     @#CONTRL,$REG0
046336  005037  001564                    CLR     $REG1           ;GROUP NO.
046342  032700  000040                    BIT     #S1,R0          ;WHICH GROUP?
046346  001403                            BEQ     13$
046350  012737  000001  001564            MOV     #1,$REG1        ;GROUP NO
046356  010337  001566            13$:    MOV     R3,$REG2        ;TEST DATA ADDRESS
046362  104073                            ERROR   73              ;PREVIOUSLY CACHED TEST-DATA
                                                                  ;WAS REFERENCED BUT IT
                                                                  ;WAS NOT A HIT.
                                                                  ;POSSIBLE FAULT: CACHE DID NOT
                                                                  ;TURN ON AFTER HAVING TURNED
                                                                  ;OFF (WHEN A CACHE FLUSH
                                                                  ;WAS DONE WITH IVSS SET).

046364  062703  000002            11$:    ADD     #2,R3
046370  077426                            SOB     R4,10$          ;DONE?
046372  052700  000014                    BIS     #MOM1,R0
046376  052701  000014                    BIS     #MOM1,R1

046402  012704  020000                    MOV     #VSIU,R4
046406  074402                            XOR     R4,R2           ;DONE VALID STORE B?
046410  001402                            BEQ     14$
046412  000137  046000                    JMP     IVFCA
046416  032700  000040            14$:    BIT     #S1,R0          ;YES, DONE GROUP 1?
046422  001007                            BNE     TST50           ;;YES,EXIT

046424  012700  000054                    MOV     #S1MOM1,R0      ;CCR MASKS FOR GROUP 1 TESTING
046430  012701  000034                    MOV     #S0MOM1,R1
046434  005002                            CLR     R2              ;BIT MASK FOR VALID STORE
046436  000137  046000                    JMP     IVFCA           ;GO TEST GROUP 1



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

```
                                               ;THIS TEST CHECKS THAT THE CACHE TURNS OFF AND FORCES
                                               ;ALL REFERENCES TO THE MAIN MEMORY WHEN BACK-TO-BACK
                                               ;CACHE FLUSHES ARE DONE. WHEN A CACHE FLUSH IS INITIATED
                                               ;WHILE THE PREVIOUS ONE IS IN PROGRESS, IT IS KNOWN
                                               ;AS BACK-TO-BACK FLUSH.
                                               ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                       ;;*************************************************************
046442  000004                         TST50:  SCOPE
046444  005037  177746                         CLR     @#CONTRL
046450  032737  010000  177746  1$:            BIT     #VCIP,@#CONTRL
046456  001374                                 BNE     1$
046460  012701  177774                         MOV     #-4,R1
046464  012700  000007                         MOV     #7,R0
046470  012737  000400  177746                 MOV     #FCAC,@#CONTRL  ;FLUSH CACHE
046476  012737  000400  177746                 MOV     #FCAC,@#CONTRL  ;AGAIN FLUSH THE CACHE. SINCE
                                                                       ;PREVIOUS FLUSH IS STILL IN
046504  005201                          2$:    INC     R1              ;PROGRESS CACHE SHOULD BE
046506  001410                                 BEQ     3$              ;TURNED OFF
046510  030037  177752                         BIT     R0,@#HITMIS     ;CHECK THAT THE LAST THREE REFERENCES
046514  001773                                 BEQ     2$              ;WERE MISSES
046516  013702  177752                         MOV     @#HITMIS,R2
046522  010237  001562                         MOV     R2,$REG0
046526  104115                                 ERROR   115             ;CACHE DID NOT TURN OFF ON
                                                                       ;PERFORMING A BACK-TO-BACK
                                                                       ;FLUSH. FOR THE PERIOD OF TIME
                                                                       ;THAT THE FLUSH IS BEING DONE
                                                                       ;AND THE CACHE IS OFF, ALL
                                                                       ;REFERENCES SHOULD BE FORCED
                                                                       ;TO MAIN MEMORY (MISSES).
046530                                  3$:                            ;EXIT



                                       ;;*************************************************************
                                       ;*TEST 51        CHECK CACHE-BYPASS
                                               ;THIS TEST CHECKS THE CACHE BYPASS FUNCTION.  WHEN THE
                                               ;'BYPASS CACHE'' IS SET IN THE CACHE CONTROL REGISTER
                                               ;ALL REFERENCES ARE FORCED TO MAIN MEMORY. IF A
                                               ;READ OR WRITE HIT OCCURS THAT LOCATION IS INVAL-
                                               ;-IDATED IN THE TAG STORE.
                                               ;FIRST, THE TEST CODE IS MADE HIT IN GROUP 1 BY
                                               ;FORCE-REPLACING GROUP 1. THEN THE TEST-DATA IS MADE
                                               ;HIT IN GROUP 0. CACHE-BYPASS IS SET AND THE TEST
                                               ;DATA (WHICH HAS BEEN CACHED IN GROUP 0) IS
                                               ;REFERENCED.  THE REFERENCES ARE CHECKED FOR MISSES
                                               ;(THE TEST-DATA INSIDE THE CACHE GROUP-0 SHOULD
                                               ;HAVE BEEN INVALIDATED WHEN REFERENCES WERE
                                               ;MADE WITH CACHE-BYPASS SET.)
                                               ;THE ENTIRE TEST IS REPEATED, SELECTING THE
                                               ;OTHER VALID STORE AND THEN WITH TEST-DATA IN
                                               ;GROUP 1.
                                               ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                       ;;*************************************************************
046530  000004                         TST51:  SCOPE
046532  005002                         CBP:    CLR     R2              ;BIT MASK FOR VSIU
046534  012700  000034                         MOV     #S0M0M1,R0
046540  012701  000054                         MOV     #S1M0M1,R1
```

```
046544  050200                          CBPA:   BIS     R2,R0
046546  050201                                  BIS     R2,R1
046550  010237  177746                          MOV     R2,@#CONTRL
046554  032737  010000  177746          1$:     BIT     #VCIP,@#CONTRL  ;SELECT VSIU
046562  001374                                  BNE     1$

046564  012703  046532                          MOV     #CBP,R3         ;MAKE TEST-CODE HIT IN THE
046570  012704  001000                          MOV     #1000,R4        ;GROUP NOT BEING TESTED
046574  010037  177746                  2$:     MOV     R0,@#CONTRL
046600  005763  002000                          TST     2000(R3)
046604  010137  177746                          MOV     R1,@#CONTRL
046610  005723                                  TST     (R3)+
046612  077410                                  SOB     R4,2$

046614  042700  000014                          BIC     #MOM1,R0
046620  042701  000014                          BIC     #MOM1,R1

046624  012703  116310                          MOV     #TSTDAT,R3      ;MAKE TEST-DATA HIT IN THE
046630  012704  001000                          MOV     #1000,R4        ;GROUP BEING TESTED
046634  010037  177746                          MOV     R0,@#CONTRL
046640  005723                          3$:     TST     (R3)+
046642  077402                                  SOB     R4,3$
046644  010137  177746                          MOV     R1,@#CONTRL     ;FORCE REPLACE IN THE GROUP NOT
                                                                        ;BEING TESTED
046650  052737  001000  177746                  BIS     #UCB,@#CONTRL   ;UNCONDITIONED CACHE BY-PASS

046656  012703  116310                          MOV     #TSTDAT,R3
046662  012704  001000                          MOV     #1000,R4        ;REFERENCE THE CACHED-TEST-DATA
046666  005713                          4$:     TST     (R3)            ;THE GROUP BEING TESTED
046670  032737  000010  177752                  BIT     #10,@#HITMIS    ;MISS?
046676  001416                                  BEQ     5$              ;YES

046700  013737  177746  001562                  MOV     @#CONTRL,$REG0
046706  005037  001564                          CLR     $REG1           ;GROUP NO.
046712  032700  000040                          BIT     #S1,R0          ;WHICH GROUP?
046716  001403                                  BEQ     8$
046720  012737  000001  001564                  MOV     #1,$REG1        ;GROUP NO
046726  010337  001566                  8$:     MOV     R3,$REG2        ;TEST DATA ADDRESS
046732  104074                                  ERROR   74              ;TEST-DATA-REFERENCE WAS NOT
                                                                        ;A MISS.  TEST-DATA WAS PREVIOUSLY
                                                                        ;CACHED IN THE GROUP BEING
                                                                        ;TESTED.  THEN IT WAS REFERENCED
                                                                        ;WITH CACHE BY-PASS SET.  IT
                                                                        ;SHOULD HAVE BEEN A MISS.
                                                                        ;PROBABLE FAILURE : A MISS IS
                                                                        ;NOT BEING FORCED WHEN CACHE
                                                                        ;BYPASS IS SET

046734  062703  000002                  5$:     ADD     #2,R3
046740  077426                                  SOB     R4,4$           ;DONE?

046742  042737  001000  177746                  BIC     #UCB,@#CONTRL   ;CLEAR CACHE BYPASS
046750  012703  116310                          MOV     #TSTDAT,R3      ;REFERENCE THE TEST-DATA AGAIN
046754  012704  000400                          MOV     #400,R4 ;IT SHOULD BE A MISS
046760  005713                          6$:     TST     (R3)
046762  032737  000010  177752                  BIT     #10,@#HITMIS    ;MISS?
```

```
046770  001416                                  BEQ     7$                      ;YES

046772  013737  177746  001562                  MOV     @#CONTRL,$REG0
047000  005037  001564                          CLR     $REG1                   ;GROUP NO.
047004  032700  000040                          BIT     #S1,R0                  ;WHICH GROUP?
047010  001403                                  BEQ     9$
047012  012737  000001  001564                  MOV     #1.$REG1                ;GROUP NO
047020  010337  001566                  9$:     MOV     R3,$REG2                ;TEST DATA ADDRESS
047024  104075                                  ERROR   75                      ;TEST-DATA-REFERENCE WAS NOT
                                                                                ;A MISS.  TEST-DATA WAS PREVIOUSLY
                                                                                ;CACHED IN THE GROUP BEING
                                                                                ;TESTED.  THEN IT WAS INVALIDATED
                                                                                ;BY REFERENCING IT WHILE
                                                                                ;CACHE-BYPASS WAS SET.  THEN
                                                                                ;CACHE-BYPASS WAS CLEARED AND
                                                                                ;THE TEST DATA WAS REFERENCED
                                                                                ;AGAIN TO MAKE SURE IT WAS
                                                                                ;INVALIDATED.
047026  062703  000004                  7$:     ADD     #4,R3                   ;PROBABLE FAILURE - CACHE-BYPASSS
047032  077426                                  SOB     R4,6$                   ;DOES NOT INVALIDATE DATE
                                                                                ;THAT IS A HIT INSIDE THE
047034  052700  000014                          BIS     #MOM1,R0                ;CACHE
047040  052701  000014                          BIS     #MOM1,R1
047044  012704  020000                          MOV     #VSIU,R4
047050  074402                                  XOR     R4,R2   ;DONE BOTH VALID STORES?
047052  001234                                  BNE     CBPA                    ;NO
047054  032700  000040                          BIT     #S1,R0                  ;TESTED GROUP 1
047060  001005                                  BNE     TST52                   ;;EXIT

047062  012700  000054                          MOV     #S1MOM1,R0              ;SET UP FOR TESTING GROUP 1
047066  012701  000034                          MOV     #S0MOM1,R1
047072  000624                                  BR      CBPA



                                        ;;*****************************************************************
                                        ;*TEST 52        CHECK CACHE IS BYPASSED ON ASRB OPERAND
                                                ;THIS TEST CHECKS THAT THE CACHE IS BYPASSED ON THE
                                                ;OPERAND OF THE ASRB INSTRUCTION AND ALSO THE OPERAND
                                                ;IS INVALIDATED.  TEST-CODE (INCLDING THE OPERAND
                                                ;OF THE ASRB) IS MADE HIT IN GROUP 1.  THEN
                                                ;ASRB INSTRUCTION IS EXECUTED ON THE CACHED
                                                ;OPERAND.  IT IS CHECKED IF THE REFERENCE TO THE
                                                ;BYTE-OPERAND WAS A MISS.  THEN THE SAME OPERAND
                                                ;REFERENCED USING AN ORDINARY (NON-BYPASSING)
                                                ;INSTRUCTED.  AGAIN, THE REFERENCE IS CHECKED FOR
                                                ;A MISS.
                                                ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                        ;;*****************************************************************
047074  000004                          TST52:  SCOPE

047076  012703  047076                  ASRBCB: MOV     #ASRBCB,R3
047102  012704  001000                          MOV     #1000,R4        ;MAKE TEST-CODE HIT IN GROUP
047106  012737  000034  177746          1$:     MOV     #S0MOM1,@#CONTRL;1
047114  005763  002000                          TST     2000(R3)
047120  012737  000054  177746                  MOV     #S1MOM1,@#CONTRL
```

```
047126  005723                              TST     (R3)+
047130  077412                              SOB     R4,1$
047132  042737  000014  177746              BIC     #MOM1,@#CONTRL
                                                                    ;EXECUTE AN ASRB AND REFERENCE
047140  106237  047242                      ASRB    @#ASLOC         ;THE TEST LOCATION
047144  032737  000010  177752              BIT     #10,@#HITMIS
047152  001412                              BEQ     2$

047154  013737  177746  001562              MOV     @#CONTRL,$REG0
047162  012737  000001  001564              MOV     #1,$REG1                ;GROUP NO.
047170  012737  047242  001566              MOV     #ASLOC,$REG2    ;TEST DATA ADDRESS
047176  104076                              ERROR   76              ;PREVIOUSLY CACHED TEST-LOCATION
                                                                    ;WHEN REFERENCED USING AN
                                                                    ;ASRB INSTRUCTION WAS NOT
                                                                    ;A MISS.
                                                                    ;PROBABLE FAILURE = ASRB DOES
                                                                    ;NOT FORCE OPERAND-REFERENCE
                                                                    ;TO THE MAIN MEMORY

047200  005737  047242              2$:     TST     @#ASLOC         ;REFERENCE THE TEST-LOCATION
047204  032737  000010  177752              BIT     #10,@#HITMIS    ;MISS?
047212  001414                              BEQ     TST53           ;;EXIT

047214  013737  177746  001562              MOV     @#CONTRL,$REG0
047222  012737  000001  001564              MOV     #1,$REG1        ;GROUP NO.
047230  012737  047242  001566              MOV     #ASLOC,$REG2    ;TEST DATA ADDRESS
047236  104077                              ERROR   77              ;BYTE-OPERAND (OF ASRB) WAS
                                                                    ;NOT INVALIDATED WHEN ASRB
                                                                    ;WAS EXECUTED ON A CACHED
                                                                    ;LOCATION
047240  000401                              BR      TST53           ;;EXIT

047242  000000                      ASLOC:  .WORD   0



                                    ;;*******************************************************************
                                    ;*TEST 53           CHECK CACHE VALID STORE PARITY CHECKER
                                            ;THIS TEST FORCES VALID STORE PARITY ERROR IN THE FOUR
                                            ;VALID STORES AND CHECKS THE PARITY CHECKERS.
                                            ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                    ;;*******************************************************************
047244  000004                      TST53:  SCOPE
047246  013700  177744                      MOV     @#MSER,R0
047252  010037  177744                      MOV     R0,@#MSER
047256  005002                      CVSPE:  CLR     R2              ;BIT MASK FOR VSIU
047260  012704  000034                      MOV     #S0M0M1,R4      ;SET UP BIT MASKS TO CHECK
047264  012705  000054                      MOV     #S1M0M1,R5      ;GROUP 1 FIRST
047270  050204                      CVSPEA: BIS     R2,R4
047272  050205                              BIS     R2,R5
047274  010237  177746                      MOV     R2,@#CONTRL
047300  032737  010000  177746         1$:  BIT     #VCIP,@#CONTRL
047306  001374                              BNE     1$

047310  010437  177746                      MOV     R4,@#CONTRL
047314  005737  051370                      TST     @#2$+2000
```

```

047320 010537 177746                MOV     R5,@#CONTRL     ;MAKE 'NOP'' HIT IN GROUP BEING
047324 005737 047370                TST     @#2$            ;TESTED

047    032704 000020                BIT     #S0,R4          ;TESTING GROUP  1?
047334 001004                       BNE     13$             ;YES
047336 042737 000004 177746         BIC     #M0,@#CONTRL    ;TESTING GROUP 0,FORCE MISS GROUP 1
047344 000403                       BR      14$
047346 042737 000010 177746 13$:    BIC     #M1,@#CONTRL    ;TESTING GROUP 1,FORCE MISS GROUP 0

047354 012737 047412 000114 14$:    MOV     #3$,@#114       ;SETUP PARITY ERROR TRAP VECTOR

047362 052737 002000 177746         BIS     #FVPE,@#CONTRL  ;FORCE VALID STORE PARITY ERROR

047370 000240                2$:    NOP                     ;REFERENCE OF THIS INSTRUCTION
                                                            ;WILL FORCE A VALID STORE
                                                            ;PARITY ERROR. TRAP TO 114
                                                            ;SHOULD OCCUR.
047372 042737 002000 177746         BIC     #FVPE,@#CONTRL  ;USER FVPE IF STILL SET
047400 013737 177746 001562         MOV     @#CONTRL,$REG0
047406 104103                       ERROR   103             ;VALID STORE PARITY ERROR WAS
                                                            ;FORCED BY SETTINGS FVPE IN
                                                            ;CCR AND MAKING A REFERENCE
                                                            ;TO A  LOC THAT WAS MADE
                                                            ;A HIT.  PARITY ERROR TRAP
                                                            ;DID NOT OCCUR ON DETECTING
                                                            ;THAT PARITY ERROR IN VALID
047410 000411                       BR      6$              ;STORE

047412                       3$:                            ;ENTER HERE IF A PARITY ERROR
                                                            ;TRAP OCCURS AS EXPECTED
047412 022626                       CMP     (SP)+, (SP)+    ;POP THE STACK
047414 032737 002000 177746  5$:    BIT     #FVPE,@#CONTRL  ;FVPE CLEARED AFTER PARITY ERROR?
047422 001404                       BEQ     6$
047424 013737 177746 001562         MOV     @#CONTRL,$REG0
047432 104105                       ERROR   105             ;FVPE DID NOT GET CLEARED
                                                            ;AFTER VALID SOTRE PARITY ERROR
                                                            ;OCCURED

047434 032737 100000 177746  6$:    BIT     #VSPE,@#CONTRL  ;DID-VALID-STORE-PARITY-ERROR SET?
047442 001004                       BNE     7$
047444 013737 177746 001562         MOV     @#CONTRL,$REG0
047452 104106                       ERROR   106             ;VALID-STORE-PARITY-ERROR BIT
                                                            ;DID NOT SET IN CCR WHEN
                                                            ;PARITY ERROR FROM V-STORE)
047454 005003                7$:    CLR     R3
                                                            ;WAS FORCED
047456 032705 000040                BIT     #S1,R5          ;TESTING GROUP 1?
047462 001003                       BNE     8$              ;YES
047464 012700 000020                MOV     #BIT4,R0        ;SET BIT MASK FOR GROUP 0
047470 000402                       BR      9$
047472 012700 000040         8$:    MOV     #BIT5,R0        ;SET BIT MASK FOR GROUP 1
047476 013701 '77744         9$:    MOV     @#MSER,R1       ;GROUP IN WHICH THE PARITY ERROR
047502 042701 177716                BIC     #^CBIT4+^CBIT5,R1;OCCURED) WAS SET IN MEMORY
047506 020100                       CMP     R1,R0           ;SYSTEM ERROR REGISTER
047510 001413                       BEQ     10$
```

```
047512  013737  177746  001562             MOV     @#CONTRL,$REG0
047520  013737  177744  001564             MOV     @#MSER,$REG1
047526  010337  001566                     MOV     R3,$REG2          ;GROUP NO. BEING TESTED
047532  010037  001570                     MOV     R0,$REG3          ;EXPECTED BITS (4,5)IN MEM SY
                                                                     ;ERROR REGISTER
047536  104107                             ERROR   107               ;PROPER BITS (ADDRESS MEMORY)
                                                                     ;PARITY ERROR.4,5) WERE NOT
                                                                     ;SET IN MEM-SYS.ERROR
                                                                     ;REGISTER WHEN VALID STORE
                                                                     ;PARITY ERROR WAS FORCED.

047540  013703  177746             10$:    MOV     @#CONTRL,R3       ;DID VSIU BIT SWITCH AFTER
047544  074203                             XOR     R2,R3             ;PARITY ERROR TRAP?
047546  032703  020000                     BIT     #VSIU,R3
047552  001404                             BEQ     11$               ;NO, OK
047554  013737  177746  001562             MOV     @#CONTRL,$REG0
047562  104110                             ERROR   110               ;VSIU  SWITCHED WHEN A VALID
                                                                     ;STORE PARITY ERROR WAS
                                                                     ;FORCED.  IT SHOULD NOT.
047564  013701  177744             11$:    MOV     @#MSER,R1         ;CLEAR MEMORY SYSTEM ERROR
047570  010137  177744                     MOV     R1,@#MSER         ;REGISTER
047574  005737  177744                     TST     @#MSER            ;CLEARED?
047600  001404                             BEQ     12$
047602  013737  177744  001562             MOV     @#MSER,$REG0
047610  104111                             ERROR   111               ;MEMORY SYSTEM ERROR REGISTER
                                                                     ;WOULD NOT BE CLEARED BY
                                                                     ;WRITING ITS CONTENTS BACK
                                                                     ;INTO ITSELF.  NOTE PREVIOUSLY
                                                                     ;A VALID STORE PARITY ERROR
                                                                     ;OCCURED THIS SETTINGS ?
                                                                     ;BIT 4 OR 5 IN IT.
047612  012700  020000             12$:    MOV     #VSIU,R0
047616  074002                             XOR     R0,R2             ;DONE VALID STORE B?
047620  001223                             BNE     CVSPEA            ;GO CHECK VALID STORE B

047622  032705  000020                     BIT     #S0,R5            ;CHECKED GROUP 0?
047626  001007                             BNE     TST54             ;;EXIT

047630  012704  000054                     MOV     #S1M0M1,R4        ;SET UP BIT MASKS TO CHECK
047634  012705  000034                     MOV     #S0M0M1,R5        ;GROUP 0
047640  005002                             CLR     R2                ;BUT MASK FOR VALID STORE
047642  000137  047270                     JMP     CVSPEA
                                   ;;***********************************************************
                                   ;*TEST 54        CHECK THAT CACHE-MISS OCCURS ON A VALID STORE PARITY ERROR
                                           ;THIS TEST FORCES A VALID STORE PARITY ERROR AND CHECKS
                                           ;THAT A MISS OCCURS ON THE REFERENCE THAT CAUSED
                                           ;THE PARITY ERROR.  THE CACHE LOCATION THAT GAVE THE
                                           ;PARITY ERROR IS INVALIDATED AND A SLOW CYCLE IS
                                           ;PERFORMED TO THE MAIN MEMORY.  THIS TEST IS
                                           ;PERFORMED WITH THE 'DISABLE TRAPS'' BIT OF THE
                                           ;CACHE CONTROL REGISTER SET, THUS A PARITY ERROR
                                           ;TRAP WILL NOT OCCUR.  THIS IS DONE SO THAT THE
                                           ;HIT-MISS REGISTER CAN BE READ WITHOUT LOSING
                                           ;THE INFORMATION CONTAINED IN IT.
                                           ;THIS TEST WILL ONLY BE EXECUTED ON A KB-11CM,E, OR EM
                                   ;;***********************************************************
```

```
047646  000004                    TST54:  SCOPE
047650  005002                    VSCM:   CLR     R2                      ;BIT MASK FOR VSIU
                                                                            ;TRAPS
047652  012704  000034                    MOV     #S0M0M1,R4              ;SET BIT MASKS TO CHECK
047656  012705  000054                    MOV     #S1M0M1,R5              ;GROUP 1 FIRST
047662  050204                    VSCMA:  BIS     R2,R4
047664  050205                            BIS     R2,R5
047666  010237  177746                    MOV     R2,@#CONTRL
047672  032737  010000  177746   1$:      BIT     #VCIP,@#CONTRL
047700  001374                            BNE     1$

047702  010437  177746                    MOV     R4,@#CONTRL             ;MAKE 'NOP'' LIST IN GROUP
047706  005737  051736                    TST     @#2$+2000               ;BEING TESTED
047712  010537  177746                    MOV     R5,@#CONTRL
047716  005737  047736                    TST     @#2$
047722  042737  000014  177746            BIC     #M0M1,@#CONTRL
047730  052737  002001  177746            BIS     #FVPE+DT,@#CONTRL       ;FORCE VALID STORE PARITY ERROR

047736  000240                   2$:      NOP                             ;REFERENCE OF THIS INSTRUCTION
                                                                          ;WILL FORCE A VALID STORE
                                                                          ;PARITY ERROR

047740  032737  000010  177752            BIT     #10,@#HITMIS            ;CHECK THAT THE REFERENCE
047746  001407                            BEQ     3$                      ;WHICH CAUSED THE V-STORE
047750  013737  177746  001562            MOV     @#CONTRL,$REG0          ;PARITY ERROR WAS A MISS
047756  013737  177744  001564            MOV     @#MSER,$REG1            ;TEXT-DATA REFERENCE WHICH
047764  104104                            ERROR   104                     ;CAUSED A PARITY ERROR
                                                                          ;(IN THE VALID STORE)
                                                                          ;SHOULD HAVE BEEEN A MISS-
                                                                          ;IT WAS NOT.

047766  032737  100000  177746   3$:      BIT     #VSPE,@#CONTRL          ;DID VALID STORE PARITY ERROR
                                                                            ;SET?
047774  001004                            BNE     4$
047776  013737  177746  001562            MOV     @#CONTRL,$REG0
050004  104106                            ERROR   106                     ;VALID STORE PARITY ERROR BIT
                                                                          ;DID NOT SET IN CCR WHEN
                                                                          ;PARITY ERROR FROM (V-STORE)
                                                                          ;WAS FORCED

050006  052737  100000  177746   4$:      BIS     #VSPE,@#CONTRL          ;CLEAR VSPE
050014  032737  100000  177746            BIT     #VSPE,@#CONTRL          ;CHECK
050022  001404                            BEQ     5$
050024  013737  177746  001562            MOV     @#CONTRL,$REG0
050032  104112                            ERROR   112                     ;VALID STORE PARITY ERROR
                                                                          ;BIT COULD NOT BE CLEARED IN CCR

050034  013737  177744  177744   5$:      MOV     @#MSER,@#MSER           ;CLEAR MEMORY SYSTEM ERROR
                                                                            ;REGISTER
050042  012700  020000                    MOV     #VSIU,R0
050046  074002                            XOR     R0,R2                   ;DONE VALID STORE B?
050050  001304                            BNE     VSCMA                   ;GO CHECK V-STORE B
050052  032705  000020                    BIT     #S0,R5                  ;CHECKED GROUP 0
050056  001007                            BNE     6$

050060  012704  000054                    MOV     #S1M0M1,R4              ;SET UP BIT MASKS ITS CHECK
```

```
050064  012705  000034                          MOV     #SOMOM1,R5              ;GROUP 0
050070  005002                                  CLR     R2
050072  000137  047662                          JMP     VSCMA

050076  005037  177746                  6$:     CLR     @#CONTRL
050102                                  KT:             .


                                        ;;*************************************************************
                                        ;*TEST 55       CHECK BYP ON KERNEL PAGE BITS
                                                ;THIS TEST IS EXECUTED ONLY ON KB11-E,KB11-EM,AND MODIFIED KB11-B/C (KB11CM)
                                        ;;*************************************************************
050102  000004                          TST55:  SCOPE
050104  012737  000012  001702                  MOV     #12,$TIMES      ;;DO 12 ITERATIONS
050112  105737  001750                          TSTB    KB11CM
050116  001003                                  BNE     5$              ;BR IF MOIFIED 11/70 (KB11CM)
050120  005737  001746                          TST     KB11E           ;IS IT A KB11-E OR KB11-EM?
050124  001444                                  BEQ     TST56           ;;
050126  012700  172300                  5$:     MOV     #KIPDR0,R0      ;POINT TO KIPDR0
050132  005010                          4$:     CLR     (R0)            ;CLEAR KIPDR
050134  032710  100000                          BIT     #BYP,(R0)       ;DID BYP CLEAR?
050140  001405                                  BEQ     1$              ;BRANCH IF YES
050142  010037  001562                          MOV     R0,$REG0
050146  011037  001564                          MOV     (R0),$REG1
050152  104123                                  ERROR   123             ;BYP STUCK SET

050154  052710  100000                  1$:     BIS     #BYP,(R0)               ;SET BYP
050160  032710  100000                          BIT     #BYP,(R0)       ;IS IT SET?
050164  001005                                  BNE     2$              ;BRANCH IF YES
050166  010037  001562                          MOV     R0,$REG0
050172  011037  001564                          MOV     (R0),$REG1
050176  104124                                  ERROR   124             ;BYP STUCK CLEAR

050200  042710  100000                  2$:     BIC     #BYP,(R0)       ;CLEAR BYP
050204  032710  100000                          BIT     #BYP,(R0)       ;IS IT CLEAR?
050210  001405                                  BEQ     3$              ;BRANCH IF YES
050212  010037  001562                          MOV     R0,$REG0
050216  011037  001564                          MOV     (R0),$REG1
050222  104123                                  ERROR   123             ;BYP STUCK SET

050224  062700  000002                  3$:     ADD     #2,R0           ;POINT TO NEXT PDR
050230  020027  172340                          CMP     R0,#KDPDR7+2    ;ARE WE FINISHED?
050234  001336                                  BNE     4$              ;BRANCH IF NOT

                                        ;;*************************************************************
                                        ;*TFST 56       CHECK BYP ON SUPERVISOR PAGE BITS
                                                ;*THIS TEST IS EXECUTED ONLY ON KB11-E, KB11-EM, AND MODIFIED KB11-B/C (KB11CM).
                                        ;;*************************************************************
050236  000004                          TST56:  SCOPE
050240  012737  000012  001702                  MOV     #12,$TIMES      ;;DO 12 ITERATIONS
050246  105737  001750                          TSTB    KB11CM
050252  001003                                  BNE     5$              ;BR IF MOIFIED 11/70 (KB11CM)
050254  005737  001746                          TST     KB11E           ;IS IT A KB11-E OR KB11-EM?
050260  001444                                  BEQ     TST57           ;;
050262  012700  172200                  5$:     MOV     #SIPDR0,R0      ;POINT TO SIPDR0
050266  005010                          4$:     CLR     (R0)            ;CLEAR SIPDR
```

```
9085  050270  032710  100000                    BIT     #BYP,(R0)          ;DID BYP CLEAR?
9086  050274  001405                            BEQ     1$                 ;BRANCH IF YES
9087  050276  010037  001562                    MOV     R0,$REG0
9088  050302  011037  001564                    MOV     (R0),$REG1
9089  050306  104130                            ERROR   130                ;BYP STUCK SET
9090
9091  050310  052710  100000            1$:     BIS     #BYP,(R0)          ;SET BYP
9092  050314  032710  100000                    BIT     #BYP,(R0)          ;DID IT SET?
9093  050320  001005                            BNE     2$                 ;BRANCH IF YES
9094  050322  010037  001562                    MOV     R0,$REG0
9095  050326  011037  001564                    MOV     (R0),$REG1
9096  050332  104131                            ERROR   131                ;BYP STUCK CLEAR
9097
9098  050334  042710  100000            2$:     BIC     #BYP,(R0)          ;CLEAR BYP
9099  050340  032710  100000                    BIT     #BYP,(R0)          ;IS IT CLEAR?
9100  050344  001405                            BEQ     3$                 ;BRANCH IF YES
9101  050346  010037  001562                    MOV     R0,$REG0
9102  050352  011037  001564                    MOV     (R0),$REG1
9103  050356  104130                            ERROR   130                ;BYP STUCK SET
9104
9105  050360  062700  000002            3$:     ADD     #2,R0              ;POINT TO NEXT PDR
9106  050364  020027  172240                    CMP     R0,#SDPDR7+2       ;ARE WE FINISHED?
9107  050370  001336                            BNE     4$                 ;BRANCH IF NO
9108                                    ;;***************************************************************
9109                                    ;*TEST 57         CHECK BYP ON USER PAGE BITS
9110                                            ;*THIS TEST IS EXECUTED ONLY ON KB11-E, KB11-EM, AND MODIFIED KB11-B/C (KB11CM).
9111                                    ;;***************************************************************
9112  050372  000004                    TST57:  SCOPE
9113  050374  012737  000012  001702            MOV     #12,$TIMES         ;;DO 12 ITERATIONS
9114  050402  105737  001750                    TSTB    KB11CM
9115  050406  001003                            BNE     5$                 ;BR IF MOIFIED 11/70 (KB11CM)
9116  050410  005737  001746                    TST     KB11E              ;IS IT A KB11-E OR KB11-EM?
9117  050414  001444                            BEQ     TST60              ;;
9118  050416  012700  177600            5$:     MOV     #UIPDR0,R0         ;POINT TO UIPDR0
9119  050422  005010                    4$:     CLR     (R0)               ;CLEAR UIPDR
9120  050424  032710  100000                    BIT     #BYP,(R0)          ;DID BYP CLEAR?
9121  050430  001405                            BEQ     1$                 ;BRANCH IF YES
9122  050432  010037  001562                    MOV     R0,$REG0
9123  050436  011037  001564                    MOV     (R0),$REG1
9124  050442  104132                            ERROR   132                ;BYP STUCK SET
9125
9126  050444  052710  100000            1$:     BIS     #BYP,(R0)          ;SET BYP
9127  050450  032710  100000                    BIT     #BYP,(R0)          ;IS IT SET?
9128  050454  001005                            BNE     2$                 ;BRANCH IF YES
9129  050456  010037  001562                    MOV     R0,$REG0
9130  050462  011037  001564                    MOV     (R0),$REG1
9131  050466  104133                            ERROR   133                ;BYP STUCK CLEAR
9132
9133  050470  042710  100000            2$:     BIC     #BYP,(R0)          ;CLEAR BYP
9134  050474  032710  100000                    BIT     #BYP,(R0)          ;IS IT CLEAR?
9135  050500  001405                            BEQ     3$                 ;BRANCH IF YES
9136  050502  010037  001562                    MOV     R0,$REG0
9137  050506  011037  001564                    MOV     (R0),$REG1
9138  050512  104132                            ERROR   132                ;BYP STUCK SET
9139
9140  050514  062700  000002            3$      ADD     #2,R0              ;POINT TO NEXT PDR
```

```
050520  020027  177640                      CMP     R0,#UDPDR7+2       ;ARE WE FINISHED?
050524  001336                              BNE     4$                 ;BRANCH IF NO
                                    ;;*******************************************************************
                                    ;*TEST 60       CHECK CACHE BYPASS ON VIRTUAL PAGE
                                            ;*THIS TEST IS EXECUTED ONLY ON KB11-EM AND 11/74          (KB11CM)
                                    ;;*******************************************************************
050526  000004                      TST60:  SCOPE


050530  013746  177776                      MOV     @#PS,-(SP)         ;CLEAR T BIT IF SET
050534  042716  000020                      BIC     #20,(SP)
050540  012746  050546                      MOV     #1$,-(SP)
050544  000002                              RTI
050546                              1$:

050546  105737  001750                      TSTB    KB11CM
050552  001005                              BNE     VPBP
050554  105737  001747                      TSTB    KB11EM             ;IS IT A KB11-EM?
050560  001002                              BNE     VPBP               ;BR IF YES
050562  000137  051360                      JMP     VPBPE
050566  012704  100000              VPBP:   MOV     #100000,R4         ;INITIALIZE APF, PAGE PAR - 4
050572  012705  172350                      MOV     #KIPAR4,R5
050576  012703  172310                      MOV     #KIPDR4,R3
050602  005037  177746              VPBPA:  CLR     @#CONTRL
050606  032737  010000  177746      1$:     BIT     #VCIP,@#CONTRL     ;WAIT FOR VCIP TO CLEAR
050614  001374                              BNE     1$

050616  012700  050566                      MOV     #VPBP,R0           ;MAKE TEST CODE HIT IN GROUP 1
050622  012701  001000                      MOV     #1000,R1
050626  012737  000034  177746      2$:     MOV     #S0M0M1,@#CONTRL
050634  005760  002000                      TST     2000(R0)
050640  012737  000054  177746              MOV     #S1M0M1,@#CONTRL
050646  005720                              TST     (R0)+
050650  077112                              SOB     R1,2$

050652  042737  000014  177746              BIC     #M0M1,@#CONTRL
050660  012700  063014                      MOV     #63014,R0          ;MAKE TEST-DATA HIT IN
050664  012701  001000                      MOV     #1000,R1           ;GROUP 0
050670  012737  000020  177746              MOV     #S0,@#CONTRL
050676  005720                      9$:     TST     (R0)+
050700  077102                              SOB     R1,9$

050702  012737  000040  177746              MOV     #S1,@#CONTRL       ;FORCE REPLACE GROUP 1

050710  005037  172340                      CLR     @#KIPAR0           ;MAP 0-4K VIRTUAL INTO
050714  012737  077406  172300              MOV     #77406,@#KIPDR0 ;0-4K PHYSICAL (TEST PROGRAM)
050722  012737  000200  172342              MOV     #200,@#KIPAR1   ;MAP 4-8K VIRTUAL INTO
050730  012737  077406  172302              MOV     #77406,@#KIPDR1 ;4-8K PHYSICAL (TEST PROGRAM)
050736  012737  000400  172344              MOV     #400,@#KIPAR2   ;MAP 8-12K VIRTUAL TO
050744  012737  077406  172304              MOV     #77406,@#KIPDR2 ;8-12K PHYSICAL
050752  012737  000600  172346              MOV     #600,@#KIPAR3   ;MAP 12-16K VIRTUAL TO
050760  012737  077406  172306              MOV     #77406,@#KIPDR3 ;12-16K PHYSICAL
050766  012737  177600  172356              MOV     #177600,@#KIPAR7;MAP I/O PAGE THROUGH PAGE7
050774  012737  077406  172316              MOV     #77406,@#KIPDR7

                                                                       ;SET UP PAR,PDR REGISTERS TO
```

```
                                                                   ;MAP THE TEST DATA BUFFER THRU THE
                                                                   ;VITUAL PAGE BEING TESTED
051002  012702  063014                      MOV     #63014,R2      ;PHYSICAL ADDRESS
051006  010200                              MOV     R2,R0          ;COPY IT
051010  072027  177772                      ASH     #-6,R0         ;FORM THE PAF (BLOCK #)
051014  010015                              MOV     R0,(R5)        ;SET UP PAF INSIDE THE PAR
051016  012713  010406                      MOV     #10406,(R3)    ;SET UP PDR

                                                                   ;FORM THE VIRTUAL ADDRESS FOR
                                                                   ;THE TEST DATA BUFFER
051022  042702  177700                      BIC     #177700,R2     ;CLEAR APF BIT POSITIONS
051026  050402                              BIS     R4,R2          ;SET APF BITS TO POINT TO THE
051030  010200                              MOV     R2,R0          ;PAR FOR THE VIRTUAL PAGE BEING
                                                                   ;TESTED
                                                                   ;R2 CONTAINS THE VIRTUAL ADDRESS
                                                                   ;OF THE TEST DATA BUFFER
051032  012737  000020  172516              MOV     #20,@#MMR3     ;ENABLE KT - 22 BIT MODE
051040  012737  000001  177572              MOV     #1,@#MMR0

051046  012701  001000                      MOV     #1000,R1       ;COUNT
051052  005712                      3$:     TST     (R2)
051054  032737  000010  177752              BIT     #10,@#HITMIS   ;HIT?
051062  001021                              BNE     4$             ;YES
051064  013737  177746  001562              MOV     @#CONTRL,$REG0
051072  010537  001564                      MOV     R5,$REG1       ;PAR ADDRESS
051076  011537  001566                      MOV     (R5),$REG2     ;PAR CONTENTS
051102  011337  001570                      MOV     (R3),$REG3     ;PDR CONTENTS
051106  010237  001572                      MOV     R2,$REG4       ;TEST DATA ADDRESS (VA)
051112  005037  177572                      CLR     @#MMR0         ;TURN OFF MEM MAN
051116  104125                              ERROR   125            ;TEST-DATA-BUFFER WAS REFERENCED.
                                                                   ;IT WAS FOUND TO BE A MISS.
                                                                   ;SHOULD HAVE BEEN A HIT
                                                                   ;BECAUSE IT WAS MADE HIT
                                                                   ;IN GROUP 0 BEFORE REFERENCING
051120  012737  000001  177572              MOV     #1,@#MMR0              ;TURN MM BACK ON
051126  062702  000002              4$:     ADD     #2,R2
051132  077131                              SOB     R1,3$
051134  010002                              MOV     R0,R2          ;COPY VIRTUAL ADDRESS OF TEST-DATA BUFFER
051136  052713  100000                      BIS     #BYP,(R3)      ;SET BYPASS IN PDR
051142  012701  001000                      MOV     #1000,R1       ;NOW REFERENCE THE TEST LOCATIONS
                                                                   ;THAT WERE MADE HITS PREVIOUSLY
051146  005710                      5$:     TST     (R0)           ;CHECK THEY ARE BEING BYPASSED
051150  032737  000010  177752              BIT     #10,@#HITMIS   ;MISS?
051156  001421                              BEQ     6$             ;YES
051160  013737  177746  001562              MOV     @#CONTRL,$REG0
051166  010537  001564                      MOV     R5,$REG1       ;PAR ADDRESS
051172  011537  001566                      MOV     (R5),$REG2     ;PAR CONTENTS
051176  011337  001570                      MOV     (R3),$REG3     ;PDR CONTENTS
051202  010037  001572                      MOV     R0,$REG4       ;TEST DATA ADDRESS (VA)
051206  005037  177572                      CLR     @#MMR0         ;TURN OFF MM
051212  104126                              ERROR   126            ;TEST DATA WAS NOT A MISS WHEN
                                                                   ;IT WAS REFERENCED WITH CACHE
                                                                   ;BYPASS (ON VIRTUAL PAGE) SET.
                                                                   ;TEST DATA WAS PREVIOUSLY MADE HIT
                                                                   ;IN GROUP 0. IT WAS MAPPED
                                                                   ;THROUGH A PAR,PDR SET
```

```
                                                              ;(BEING TESTED). BYPASS BIT WAS
                                                              ;SET IN THE PDR AND TEST-LOC
                                                              ;WAS REFERENCED. IT SHOULD HAVE
                                                              ;BEEN A MISS (BECAUSE OF BYPASS)
                                                              ;PROBABLE FAULT: SETTING CACHE
                                                              ;BYPASS IN PDR DOES NOT BYPASS
                                                              ;VIRTUAL REFERENCES MAPPED THRU
                                                              ;THAT PAGE.
051214  012737  000001  177572          MOV     #1,@#MMR0     ;TURN MM BACK ON


051222  062700  000002          6$:     ADD     #2,R0
051226  077131                          SOB     R1,5$
051230  042713  100000                  BIC     #BYP,(R3)     ;CLEAR BYPASS IN PDR
051234  012701  001000                  MOV     #1000,R1      ;REFERENCE THE TEST-DATA AND
                                                              ;MAKE SURE IT WAS INVALIDATED
                                                              ;ON PREVIOUS BYPASS
051240  005712                  7$:     TST     (R2)          ;REFERENCE TEST DATA
051242  032737  000010  177746          BIT     #10,@#CONTRL  ;MISS?
051250  001421                          BEQ     8$
051252  013737  177746  001562          MOV     @#CONTRL,$REG0
051260  010537  001564                  MOV     R5,$REG1      ;PAR ADDRESS
051264  011537  001566                  MOV     (R5),$REG2    ;PAR CONTENTS
051270  011337  001570                  MOV     (R3),$REG3    ;PDR CONTENTS
051274  010237  001572                  MOV     R2,$REG4      ;TEST DATA ADDRESS (VIRTUAL ADDRESS)
051300  005037  177572                  CLR     @#MMR0        ;TURN OFF MM
051304  104127                          ERROR   127           ;TEST-DATA REFERENCE WAS NOT
                                                              ;A MISS.  PROBABLE FAILURE:
                                                              ;PREVIOUSLY CACHED TEST DATA
                                                              ;LOCATIONS WERE NOT INVALIDATED
                                                              ;(IN THE CACHE) WHEN CACHE
                                                              ;BYPASS WAS FORCED ON REFERENCES THROUGH
                                                              ;THE VIRTUAL PAGE (BEING TESTED).
051306  012737  000001  177572          MOV     #1,@#MMR0     ;TURN MM BACK ON

051314  062702  000002          8$:     ADD     #2,R2         ;NEXT LOCATION
051320  077131                          SOB     R1,7$         ;DONE?


051322  005037  177572                  CLR     @#MMR0        ;DISABLE KT
051326  005037  172516                  CLR     @#MMR3
051332  062704  020000                  ADD     #20000,R4     ;INITIALIZE APF FIELD MASK FOR THE
051336  062705  000002                  ADD     #2,R5         ;NEXT PAR TO BE TESTED
051342  062703  000002                  ADD     #2,R3         ;NEXT PDR TO BE TESTED

051346  020327  172316                  CMP     R3,#KIPDR0+16 ;DONE TESTING EVERY PDR?
051352  001402                          BEQ     VPBPE
051354  000137  050602                  JMP     VPBPA
051360                          VPBPE:
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;;*INCREMENT THE PASS NUMBER ($PASS)
                                ;;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                ;;*TYPE 'END PASS #XXXXX'' (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;;*IF THERES A MONITOR GO TO IT
```

```
                                         ;*IF THERE ISN'T JUMP TO LOOP

051360                                   $EOP:
051360  000004                                   SCOPE
051362  005037  001502                           CLR     $TSTNM          ;;ZERO THE TEST NUMBER
051366  005037  001702                           CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
051372  005237  001724                           INC     $PASS           ;;INCREMENT THE PASS NUMBER
051376  042737  100000  001724                   BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
051404  005327                                   DEC     (PC)+           ;;LOOP?
051406  000001                           $EOPCT: .WORD   1
051410  003031                                   BGT     $DOAGN          ;;YES
051412  012737                                   MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
051414  000001                           $ENDCT: .WORD   1
051416  051406                                   $EOPCT
051420  104401  051503                           TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
051424  013746  001724                           MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
051430  104405                                   TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
051432  104401  051500                           TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
051436  013700  000042                   $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
051442  001414                                   BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
051444  012703  125252                           MOV     #125252,R3
051450  004737  056106                           JSR     PC,CHAINQ
051454  013700  000042                           MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
051460  001405                                   BEQ     $DOAGN          ;;RETURN ADDRESS
051462  000005                                   RESET                   ;;CLEAR THE WORLD
051464  004710                           $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
051466  000240                                   NOP                     ;;SAVE ROOM
051470  000240                                   NOP                     ;;FOR
051472  000240                                   NOP                     ;;ACT11
051474                                   $DOAGN:
051474  000137                                   JMP     @(PC)+          ;;RETURN
051476  005306                           $RTNAD: .WORD   LOOP
051500     377     377     000           $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
051503     015  042412  042116           $ENDMG: .ASCIZ  <15><12>/END PASS #/
051510  050040  051501  020123
051516  000043

                                         .SBTTL  SCOPE HANDLER ROUTINE

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

051520                                   $SCOPE:
051520  104407                                   CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
051522  032777  040000  130010           1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
051530  001117                                   BNE     $OVER           ;;YES IF SW14=1
                                         ;#####START OF CODE FOR THE XOR TESTER#####
```

```
051532  000416                          $XTSTR: BR      6$                      ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                                ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
051534  013746  000004                          MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
051540  012737  051560  000004                  MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
051546  005737  177060                          TST     @#177060                ;;TIME OUT ON XOR?
051552  012637  000004                          MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
051556  000466                                  BR      $SVLAD                  ;;GO TO THE NEXT TEST
051560  022626                          5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
051562  012637  000004                          MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
051566  000426                                  BR      7$                      ;;LOOP ON THE PRESENT TEST
051570                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
051570  032777  000400  127742                  BIT     #BIT08,@SWR             ;;LOOP ON SPEC. TEST?
051576  001407                                  BEQ     2$                      ;;BR IF NO
051600  017746  127734                          MOV     @SWR,-(SP)              ;;SET DESIRED TEST NUM. FROM SWR
051604  042716  000200                          BIC     #$SWRMK,(SP)            ;;STRIP AWAY UNDESIRED BITS
051610  122637  001502                          CMPB    (SP)+,$TSTNM            ;;ON THE RIGHT TEST?
051614  001465                                  BEQ     $OVER                   ;;BR IF YES
051616  105737  001503                  2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
051622  001421                                  BEQ     3$                      ;;BR IF NO
051624  123737  001515  001503                  CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
051632  101015                                  BHI     3$                      ;;BR IF NO
051634  032777  001000  127676                  BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
051642  001404                                  BEQ     4$                      ;;BR IF NO
051644  013737  001510  001506  7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
051652  000446                                  BR      $OVER
051654  105037  001503                  4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
051660  005037  001702                          CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
051664  000415                                  BR      1$                      ;;ESCAPE TO THE NEXT TEST
051666  032777  004000  127644  3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
051674  001011                                  BNE     1$                      ;;BR IF YES
051676  005737  001724                          TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
051702  001406                                  BEQ     1$                      ;;      INHIBIT ITERATIONS
051704  005237  001504                          INC     $ICNT                   ;;INCREMENT ITERATION COUNT
051710  023737  001702  001504                  CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
051716  002024                                  BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
051720  012737  000001  001504  1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
051726  013737  052004  001702          MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
051734  105237  001502          $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
051740  113737  001502  001722          MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
051746  011637  001506                          MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
051752  011637  001510                          MOV     (SP),$LPERR             ;;SAVE ERROR LOOP ADDRESS
051756  005037  001704                          CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
051762  112737  000001  001515                  MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
051770  013777  001502  127544  $OVER:  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
051776  013716  001506                          MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
052002  000002                                  RTI                             ;;FIXES PS
052004  000001                          $MXCNT: 1                               ;;MAX. NUMBER OF ITERATIONS

                                        .SBTTL  ERROR HANDLER ROUTINE

                                        ;;***************************************************************
                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO ERTYPE ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
```

```
                                      ;*SW13=1          INHIBIT ERROR TYPEOUTS
                                      ;*SW10=1          BELL ON ERROR
                                      ;*SW09=1          LOOP ON ERROR
                                      ;*CALL
                                      ;*        ERROR   N          ;;ERROR=EMT AND N-ERROR ITEM NUMBER

 052006                               $ERROR:
 052006  104407                                 CKSWR                     ;;TEST FOR CHANGE IN SOFT-SWR
 052010  105237  001503               7$:       INCB    $ERFLG            ;;SET THE ERROR FLAG
 052014  001775                                 BEQ     7$                ;;DON'T LET THE FLAG GO TO ZERO
 052016  013777  001502  127516                 MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER AND ERROR FLAG
 052024  032777  002000  127506                 BIT     #BIT10,@SWR       ;;BELL ON ERROR?
 052032  001402                                 BEQ     1$                ;;NO - SKIP
 052034  104401  001706                         TYPE    ,$BELL            ;;RING BELL
 052040  005237  001512               1$:       INC     $ERTTL            ;;COUNT THE NUMBER OF ERRORS
 052044  011637  001516                         MOV     (SP),$ERRPC       ;;GET ADDRESS OF ERROR INSTRUCTION
 052050  162737  000002  001516                 SUB     #2,$ERRPC
 052056  117737  127434  001514                 MOVB    @$ERRPC,$ITEMB    ;;STRIP AND SAVE THE ERROR ITEM CODE
 052064  032777  020000  127446                 BIT     #BIT13,@SWR       ;;SKIP TYPEOUT IF SET
 052072  001004                                 BNE     20$               ;;SKIP TYPEOUTS
 052074  004737  056354                         JSR     PC,ERTYPE         ;;GO TO USER ERROR ROUTINE
 052100  104401  001713                         TYPE    ,$CRLF
 052104                               20$:
 052104  122737  000001  001736                 CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
 052112  001007                                 BNE     2$                ;;NO,SKIP APT ERROR REPORT
 052114  113737  001514  052126                 MOVB    $ITEMB,21$        ;;SET ITEM NUMBER AS ERROR NUMBER
 052122  004737  052236                         JSR     PC,$ATY4          ;;REPORT FATAL ERROR TO APT
 052126     000                       21$:      .BYTE   0
 052127     000                                 .BYTE   0
 052130  000777                       22$:      BR      22$               ;;APT ERROR LOOP
 052132  005777  127402               2$:       TST     @SWR              ;;HALT ON ERROR
 052136  100002                                 BPL     3$                ;;SKIP IF CONTINUE
 052140  000000                                 HALT                      ;;HALT ON ERROR!
 052142  104407                                 CKSWR                     ;;TEST FOR CHANGE IN SOFT-SWR
 052144  032777  001000  127366       3$:       BIT     #BIT09,@SWR       ;;LOOP ON ERROR SWITCH SET?
 052152  001402                                 BEQ     4$                ;;BR IF NO
 052154  013716  001510                         MOV     $LPERR,(SP)       ;;FUDGE RETURN FOR LOOPING
 052160  005737  001704               4$:       TST     $ESCAPE           ;;CHECK FOR AN ESCAPE ADDRESS
 052164  001402                                 BEQ     5$                ;;BR IF NONE
 052166  013716  001704                         MOV     $ESCAPE,(SP)      ;;FUDGE RETURN ADDRESS FOR ESCAPE
 052172                               5$:
 052172  022737  051464  000042                 CMP     #$ENDAD,@#42      ;;ACT-11 AUTO-ACCEPT?
 052200  001001                                 BNE     6$                ;;BRANCH IF NO
 052202  000000                                 HALT                      ;;YES
 052204                               6$:
 052204  012737  177777  177744                 MOV     #-1,@#MEMERR
 052212  005037  177766                         CLR     @#CPUERR
 052216  000002                                 RTI


                                      .SBTTL  APT COMMUNICATIONS ROUTINE

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

 052220  112737  000001  052464       $ATY1:    MOVB    #1,$FFLG          ;;TO REPORT FATAL ERROR
 052226  112737  000001  052462       $ATY3:    MOVB    #1,$MFLG          ;;TO TYPE A MESSAGE
 052234  000403                                 BR      $ATYC
```

```
052236  112737  000001  052464  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
052244                          $ATYC:
052244  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
052246  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052250  105737  052462                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
052254  001450                          BEQ     5$              ;;IF NOT:  BR
052256  122737  000001  001736          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
052264  001031                          BNE     3$              ;;IF NOT:  BR
052266  132737  000100  001737          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
052274  001425                          BEQ     3$              ;;IF NOT:  BR
052276  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
052302  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
052310  005737  001716          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
052314  001375                          BNE     1$              ;;IF NOT:  WAIT
052316  010037  001732                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
052322  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
052324  001376                          BNE     2$
052326  163700  001732                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
052332  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
052334  010037  001734                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
052340  012737  000004  001716          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
052346  000413                          BR      5$
052350  017637  000004  052374  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
052356  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
052364  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
052370  004737  053336                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
052374  000000                  4$:     .WORD   0
052376                          5$:
052376  105737  052464          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
052402  001416                          BEQ     12$             ;;IF NOT:  BR
052404  005737  001736                  TST     $ENV            ;;RUNNING UNDER APT?
052410  001413                          BEQ     12$             ;;IF NOT:  BR
052412  005737  001716          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
052416  001375                          BNE     11$             ;;IF NOT:  WAIT
052420  017637  000004  001720          MOV     @4(SP),$FATAL   ;;GET ERROR #
052426  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
052434  005237  001716                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
052440  105037  052464          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
052444  105037  052463                  CLRB    $LFLG           ;;CLEAR LOG FLAG
052450  105037  052462                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
052454  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
052456  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
052460  000207                          RTS     PC              ;;RETURN
052462     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
052463     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
052464     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
        052466                          .EVEN
        000200                  APTSIZE=200
        000001                  APTENV=001
        000100                  APTSPOOL=100
        000040                  APTCSUP=040
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*************************************************************
                                .ENABL  LSB

```

```
                                        ;;*************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
052466  022737  000176  001540  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
052474  001074                          BNE     15$             ;;BRANCH IF NO
052476  105777  127042                  TSTB    @$TKS           ;;CHAR THERE?
052502  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
052504  117746  127036                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
052510  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
052514  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
052520  001062                          BNE     15$             ;;NO, RETURN TO USER
052522  123727  001534  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
052530  001456                          BEQ     15$             ;;BRANCH IF YES

052532  104401  053213                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
052536  104401  053220          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
052542  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
052546  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
052550  104401  053231                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
052554  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
052556  005046                          CLR     -(SP)           ;;THE NEW SWR
052560  105777  126760          7$:     TSTB    @$TKS           ;;CHAR THERE?
052564  100375                          BPL     7$              ;;IF NOT TRY AGAIN

052566  117746  126754                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
052572  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII



052576  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
052602  001005                          BNE     10$             ;;BRANCH IF NOT
052604  104401  053206                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
052610  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
052614  000757                          BR      19$             ;;LET'S TRY IT AGAIN


052616  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
052622  001022                          BNE     16$             ;;BRANCH IF NO
052624  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
052630  001403                          BEQ     11$             ;;BRANCH IF YES
052632  016677  000002  126700          MOV     2(SP),@SWR      ;;SAVE NEW SWR
052640  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
052644  104401  001713          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
052650  123727  001535  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
052656  001003                          BNE     15$             ;;BRANCH IF NOT
052660  012777  000100  126656          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
052666  000002                  15$:    RTI                     ;;RETURN
052670  004737  053550          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
052674  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
052700  002420                          BLT     18$             ;;BRANCH IF YES
052702  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
052706  003015                          BGT     18$             ;;BRANCH IF YES
052710  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
052714  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
```

```
052720 001403                      BEQ     17$                ;;BRANCH IF YES
052722 006316                      ASL     (SP)               ;;NO, SHIFT PRESENT
052724 006316                      ASL     (SP)               ;;   CHAR OVER TO MAKE
052726 006316                      ASL     (SP)               ;;   ROOM FOR NEW ONE.
052730 005266 000002        17$:   INC     2(SP)              ;;KEEP COUNT OF CHAR
052734 056616 177776               BIS     -2(SP),(SP)        ;;SET IN NEW CHAR
052740 000707                      BR      7$                 ;;GET THE NEXT ONE
052742 104401 001712        18$:   TYPE    ,$QUES             ;;TYPE ?<CR><LF>
052746 000720                      BR      20$                ;;SIMULATE CONTROL-U
                            .DSABL LSB


                            ;;*******************************************************************
                            ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                            ;*CALL:
                            ;*      RDCHR                     ;;INPUT A SINGLE CHARACTER FROM THE TTY
                            ;*      RETURN HERE               ;;CHARACTER IS ON THE STACK
                            ;*                                ;;WITH PARITY BIT STRIPPED OFF
                            ;

052750 011646               $RDCHR: MOV    (SP),-(SP)         ;;PUSH DOWN THE PC
052752 016666 000004 000002         MOV    4(SP),2(SP)        ;;SAVE THE PS
052760 105777 126560        1$:     TSTB   @$TKS              ;;WAIT FOR
052764 100375                       BPL    1$                 ;;A CHARACTER
052766 117766 126554 000004         MOVB   @$TKB,4(SP)        ;;READ THE TTY
052774 042766 177600 000004         BIC    #^C<177>,4(SP)     ;;GET RID OF JUNK IF ANY
053002 026627 000004 000023         CMP    4(SP),#23          ;;IS IT A CONTROL-S?
053010 001013                       BNE    3$                 ;;BRANCH IF NO
053012 105777 126526        2$:     TSTB   @$TKS              ;;WAIT FOR A CHARACTER
053016 100375                       BPL    2$                 ;;LOOP UNTIL ITS THERE
053020 117746 126522                MOVB   @$TKB,-(SP)        ;;GET CHARACTER
053024 042716 177600                BIC    #^C177,(SP)        ;;MAKE IT 7-BIT ASCII
053030 022627 000021                CMP    (SP)+,#21          ;;IS IT A CONTROL-Q?
053034 001366                       BNE    2$                 ;;IF NOT DISCARD IT
053036 000750                       BR     1$                 ;;YES, RESUME
053040 026627 000004 000140 3$:     CMP    4(SP),#140         ;;IS IT UPPER CASE?
053046 002407                       BLT    4$                 ;;BRANCH IF YES
053050 026627 000004 000175         CMP    4(SP),#175         ;;IS IT A SPECIAL CHAR?
053056 003003                       BGT    4$                 ;;BRANCH IF YES
053060 042766 000040 000004         BIC    #40,4(SP)          ;;MAKE IT UPPER CASE
053066 000002               4$:     RTI                       ;;GO BACK TO USER
                            ;;*******************************************************************
                            ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                            ;*CALL:
                            ;*      RDLIN                     ;;INPUT A STRING FROM THE TTY
                            ;*      RETURN HERE               ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                            ;*                                ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

053070 010346               $RDLIN: MOV    R3,-(SP)           ;;SAVE R3
053072 012703 053176        1$:     MOV    #$TTYIN,R3         ;;GET ADDRESS
053076 022703 053206        2$:     CMP    #$TTYIN+8.,R3      ;;BUFFER FULL?
053102 101405                       BLOS   4$                 ;;BR IF YES
053104 104410                       RDCHR                     ;;GO READ ONE CHARACTER FROM THE TTY
053106 112613                       MOVB   (SP)+,(R3)         ;;GET CHARACTER
053110 122713 000177        10$:    CMPB   #177,(R3)          ;;IS IT A RUBOUT
053114 001003                       BNE    3$                 ;;SKIP IF NOT
```

```
053116  104401  001712           4$:      TYPE    ,$QUES          ;;TYPE A '?'
053122  000763                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
053124  111337  053174           3$:      MOVB    (R3),9$         ;;ECHO THE CHARACTER
053130  104401  053174                    TYPE    ,9$
053134  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
053140  001356                            BNE     2$              ;;LOOP IF NOT RETURN
053142  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
053146  104401  001714                    TYPE    ,$LF            ;;TYPE A LINE FEED
053152  012603                            MOV     (SP)+,R3        ;;RESTORE R3
053154  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
053156  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
053164  012766  053176  000004            MOV     #$TTYIN,4(SP)
053172  000002                            RTI                     ;;RETURN
053174     000                   9$:      .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
053175     000                            .BYTE   0               ;;TERMINATOR
053176  000010                   $TTYIN:  .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
053206  052536  005015     000   $CNTLU:  .ASCIZ  /^U/<15><12>    ;;CONTROL 'U'
053213     136  006507  000012   $CNTLG:  .ASCIZ  /^G/<15><12>    ;;CONTROL 'G'
053220  005015  053523  020122   $MSWR:   .ASCIZ  <15><12>/SWR = /
053226  020075     000
053231     040  047040  053505   $MNEW:   .ASCIZ  /  NEW = /
053236  036440  000040


                                 .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                 ;;*****************************************************************
                                 ;*SAVE R0-R5
                                 ;*CALL:
                                 ;*       SAVREG
                                 ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                 ;*
                                 ;*TOP---(+16)
                                 ;* +2---(+18)
                                 ;* +4---R5
                                 ;* +6---R4
                                 ;* +8---R3
                                 ;*+10---R2
                                 ;*+12---R1
                                 ;*+14---R0

053242                           $SAVREG:
053242  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
053244  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
053246  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
053250  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
053252  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
053254  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
053256  016646  000022                    MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
053262  016646  000022                    MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
053266  016646  000022                    MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
053272  016646  000022                    MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
053276  000002                            RTI

                                 ;*RESTORE R0-R5
                                 ;*CALL:
                                      ;*      RESREG
053300                                $RESREG:
053300  012666  000022                        MOV     (SP)+,22(SP)      ;;RESTORE PC OF CALL
053304  012666  000022                        MOV     (SP)+,22(SP)      ;;RESTORE PS OF CALL
053310  012666  000022                        MOV     (SP)+,22(SP)      ;;RESTORE PC OF MAIN FLOW
053314  012666  000022                        MOV     (SP)+,22(SP)      ;;RESTORE PS OF MAIN FLOW
053320  012605                                MOV     (SP)+,R5          ;;POP STACK INTO R5
053322  012604                                MOV     (SP)+,R4          ;;POP STACK INTO R4
053324  012603                                MOV     (SP)+,R3          ;;POP STACK INTO R3
053326  012602                                MOV     (SP)+,R2          ;;POP STACK INTO R2
053330  012601                                MOV     (SP)+,R1          ;;POP STACK INTO R1
053332  012600                                MOV     (SP)+,R0          ;;POP STACK INTO R0
053334  000002                                RTI

                                      .SBTTL  TYPE ROUTINE

                                      ;;*****************************************************************
                                      ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                      ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                      ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                      ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                      ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                      ;*
                                      ;*CALL:
                                      ;*1) USING A TRAP INSTRUCTION
                                      ;*       TYPE    ,MESADR           ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                      ;*OR
                                      ;*       TYPE
                                      ;*       MESADR
                                      ;*

053336  105737  001557                $TYPE:  TSTB    $TPFLG            ;;IS THERE A TERMINAL?
053342  100002                                BPL     1$                ;;BR IF YES
053344  000000                                HALT                      ;;HALT HERE IF NO TERMINAL
053346  000430                                BR      3$                ;;LEAVE
053350  010046                        1$:     MOV     R0,-(SP)          ;;SAVE R0
053352  017600  000002                        MOV     @2(SP),R0         ;;GET ADDRESS OF ASCIZ STRING
053356  122737  000001  001736                CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
053364  001011                                BNE     62$               ;;NO,GO CHECK FOR APT CONSOLE
053366  132737  000100  001737                BITB    #APTSPOOL,$ENVM   ;;SPOOL MESSAGE TO APT
053374  001405                                BEQ     62$               ;;NO,GO CHECK FOR CONSOLE
053376  010037  053406                        MOV     R0,61$            ;;SETUP MESSAGE ADDRESS FOR APT
053402  004737  052226                        JSR     PC,$ATY3          ;;SPOOL MESSAGE TO APT
053406  000000                        61$:    .WORD   0                 ;;MESSAGE ADDRESS
053410  132737  000040  001737        62$:    BITB    #APTCSUP,$ENVM    ;;APT CONSOLE SUPPRESSED
053416  001003                                BNE     60$               ;;YES,SKIP TYPE OUT
053420  112046                        2$:     MOVB    (R0)+,-(SP)       ;;PUSH CHARACTER TO BE TYPED ONTO STACK
053422  001005                                BNE     4$                ;;BR IF IT ISN'T THE TERMINATOR
053424  005726                                TST     (SP)+             ;;IF TERMINATOR POP IT OFF THE STACK
053426  012600                        60$:    MOV     (SP)+,R0          ;;RESTORE R0
053430  062716  000002                3$:     ADD     #2,(SP)           ;;ADJUST RETURN PC
053434  000002                                RTI                       ;;RETURN
053436  122716  000011                4$:     CMPB    #HT,(SP)          ;;BRANCH IF <HT>
053442  001430                                BEQ     8$
053444  122716  000200                        CMPB    #CRLF,(SP)        ;;BRANCH IF NOT <CRLF>
053450  001006                                BNE     5$
```

```
053452  005726                                    TST     (SP)+             ;;POP  <CR><LF> EQUIV
053454  104401                                    TYPE                      ;;TYPE A CR AND LF
053456  001713                                    $CRLF
053460  105037  053614                            CLRB    $CHARCNT          ;;CLEAR CHARACTER COUNT
053464  000755                                    BR      2$                ;;GET NEXT CHARACTER
053466  004737  053550                  5$:       JSR     PC,$TYPEC         ;;GO TYPE THIS CHARACTER
053472  123726  001556                  6$.       CMPB    $FILLC,(SP)+      ;;IS IT TIME FOR FILLER CHARS.?
053476  001350                                    BNE     2$                ;;IF NO GO GET NEXT CHAR.
053500  013746  001554                            MOV     $NULL,-(SP)       ;;GET # OF FILLER CHARS. NEEDED
                                                                            ;;AND THE NULL CHAR.
053504  105366  000001                  7$:       DECB    1(SP)             ;;DOES A NULL NEED TO BE TYPED?
053510  002770                                    BLT     6$                ;;BR IF NO--GO POP THE NULL OFF OF STACK
053512  004737  053550                            JSR     PC,$TYPEC         ;;GO TYPE A NULL
053516  105337  053614                            DECB    $CHARCNT          ;;DO NOT COUNT AS A COUNT
053522  000770                                    BR      7$                ;;LOOP

                                        ;HORIZONTAL TAB PROCESSOR

053524  112716  000040                  8$:       MOVB    #' ,(SP)          ;;REPLACE TAB WITH SPACE
053530  004737  053550                  9$:       JSR     PC,$TYPEC         ;;TYPE A SPACE
053534  132737  000007  053614                    BITB    #7,$CHARCNT       ;;BRANCH IF NOT AT
053542  001372                                    BNE     9$                ;;TAB STOP
053544  005726                                    TST     (SP)+             ;;POP SPACE OFF STACK
053546  000724                                    BR      2$                ;;GET NEXT CHARACTER
053550  105777  125774                  $TYPEC:   TSTB    @$TPS             ;;WAIT UNTIL PRINTER IS READY
053554  100375                                    BPL     $TYPEC
053556  116677  000002  125766                    MOVB    2(SP),@$TPB       ;;LOAD CHAR TO BE TYPED INTO DATA REG.
053564  122766  000015  000002                    CMPB    #CR,2(SP)         ;;IS CHARACTER A CARRIAGE RETURN?
053572  001003                                    BNE     1$                ;;BRANCH IF NO
053574  105037  053614                            CLRB    $CHARCNT          ;;YES--CLEAR CHARACTER COUNT
053600  000406                                    BR      $TYPEX            ;;EXIT
053602  122766  000012  000002          1$:       CMPB    #LF,2(SP)         ;;IS CHARACTER A LINE FEED?
053610  001402                                    BEQ     $TYPEX            ;;BRANCH IF YES
053612  105227                                    INCB    (PC)+             ;;COUNT THE CHARACTER
053614  000000                          $CHARCNT:.WORD    0                 ;;CHARACTER COUNT STORAGE
053616  000207                          $TYPEX:   RTS     PC


                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

053620  017646  000000                  $TYPOS: MOV     @(SP),-(SP)             ;;PICKUP THE MODE
053624  116637  000001  054043                  MOVB    1(SP),$0FILL            ;;LOAD ZERO FILL SWITCH
053632  112637  054045                          MOVB    (SP)+,$0MODE+1          ;;NUMBER OF DIGITS TO TYPE
053636  062716  000002                          ADD     #2,(SP)                 ;;ADJUST RETURN ADDRESS
053642  000406                                  BR      $TYPON
053644  112737  000001  054043  $TYPOC: MOVB    #1,$0FILL               ;;SET THE ZERO FILL SWITCH
053652  112737  000006  054045                  MOVB    #6,$0MODE+1             ;;SET FOR SIX(6) DIGITS
053660  112737  000005  054042  $TYPON: MOVB    #5,$0CNT                ;;SET THE ITERATION COUNT
053666  010346                                  MOV     R3,-(SP)                ;;SAVE R3
053670  010446                                  MOV     R4,-(SP)                ;;SAVE R4
053672  010546                                  MOV     R5,-(SP)                ;;SAVE R5
053674  113704  054045                          MOVB    $0MODE+1,R4             ;;GET THE NUMBER OF DIGITS TO TYPE
053700  005404                                  NEG     R4
053702  062704  000006                          ADD     #6,R4                   ;;SUBTRACT IT FOR MAX. ALLOWED
053706  110437  054044                          MOVB    R4,$0MODE               ;;SAVE IT FOR USE
053712  113704  054043                          MOVB    $0FILL,R4               ;;GET THE ZERO FILL SWITCH
053716  016605  000012                          MOV     12(SP),R5               ;;PICKUP THE INPUT NUMBER
053722  005003                                  CLR     R3                      ;;CLEAR THE OUTPUT WORD
053724  006105                          1$:     ROL     R5                      ;;ROTATE MSB INTO ''C''
053726  000404                                  BR      3$                      ;;GO DO MSB
053730  006105                          2$:     ROL     R5                      ;;FORM THIS DIGIT
053732  006105                                  ROL     R5
053734  006105                                  ROL     R5
053736  010503                                  MOV     R5,R3
053740  006103                          3$:     ROL     R3                      ;;GET LSB OF THIS DIGIT
053742  105337  054044                          DECB    $0MODE                  ;;TYPE THIS DIGIT?
053746  100016                                  BPL     7$                      ;;BR IF NO
053750  042703  177770                          BIC     #177770,R3              ;;GET RID OF JUNK
053754  001002                                  BNE     4$                      ;;TEST FOR 0
053756  005704                                  TST     R4                      ;;SUPPRESS THIS 0?
053760  001403                                  BEQ     5$                      ;;BR IF YES
053762  005204                          4$:     INC     R4                      ;;DON'T SUPPRESS ANYMORE 0'S
053764  052703  000060                          BIS     #'0,R3                  ;;MAKE THIS DIGIT ASCII
053770  052703  000040                  5$:     BIS     #' ,R3                  ;;MAKE ASCII IF NOT ALREADY
053774  110337  054040                          MOVB    R3,8$                   ;;SAVE FOR TYPING
054000  104401  054040                          TYPE    ,8$                     ;;GO TYPE THIS DIGIT
054004  105337  054042                  7$:     DECB    $0CNT                   ;;COUNT BY 1
054010  003347                                  BGT     2$                      ;;BR IF MORE TO DO
054012  002402                                  BLT     6$                      ;;BR IF DONE
054014  005204                                  INC     R4                      ;;INSURE LAST DIGIT ISN'T A BLANK
054016  000744                                  BR      2$                      ;;GO DO THE LAST DIGIT
054020  012605                          6$:     MOV     (SP)+,R5                ;;RESTORE R5
054022  012604                                  MOV     (SP)+,R4                ;;RESTORE R4
054024  012603                                  MOV     (SP)+,R3                ;;RESTORE R3
054026  016666  000002  000004                  MOV     2(SP),4(SP)             ;;SET THE STACK FOR RETURNING
054034  012616                                  MOV     (SP)+,(SP)
054036  000002                                  RTI                             ;;RETURN
054040     000                          8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
054041     000                                  .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
```

```
054042     000                           $OCNT:   .BYTE   0                   ;;OCTAL DIGIT COUNTER
054043     000                           $OFILL:  .BYTE   0                   ;;ZERO FILL SWITCH
054044  000000                           $OMODE:  .WORD   0                   ;;NUMBER OF DIGITS TO TYPE

                                         .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                         ;;*************************************************************
                                         ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                         ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                         ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                         ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                         ;*REPLACED WITH SPACES.
                                         ;*CALL:
                                         ;*        MOV     NUM,-(SP)           ;;PUT THE BINARY NUMBER ON THE STACK
                                         ;*        TYPDS                       ;;GO TO THE ROUTINE

054046                                   $TYPDS:
054046  010046                                    MOV     R0,-(SP)            ;;PUSH R0 ON STACK
054050  010146                                    MOV     R1,-(SP)            ;;PUSH R1 ON STACK
054052  010246                                    MOV     R2,-(SP)            ;;PUSH R2 ON STACK
054054  010346                                    MOV     R3,-(SP)            ;;PUSH R3 ON STACK
054056  010546                                    MOV     R5,-(SP)            ;;PUSH R5 ON STACK
054060  012746  020200                            MOV     #20200,-(SP)        ;;SET BLANK SWITCH AND SIGN
054064  016605  000020                            MOV     20(SP),R5           ;;GET THE INPUT NUMBER
054070  100004                                    BPL     1$                  ;;BR IF INPUT IS POS.
054072  005405                                    NEG     R5                  ;;MAKE THE BINARY NUMBER POS.
054074  112766  000055  000001                    MOVB    #'-,1(SP)           ;;MAKE THE ASCII NUMBER NEG.
054102  005000                           1$:      CLR     R0                  ;;ZERO THE CONSTANTS INDEX
054104  012703  054262                            MOV     #$DBLK,R3           ;;SETUP THE OUTPUT POINTER
054110  112723  000040                            MOVB    #' ,(R3)+           ;;SET THE FIRST CHARACTER TO A BLANK
054114  005002                           2$:      CLR     R2                  ;;CLEAR THE BCD NUMBER
054116  016001  054252                            MOV     $DTBL(R0),R1        ;;GET THE CONSTANT
054122  160105                           3$:      SUB     R1,R5               ;;FORM THIS BCD DIGIT
054124  002402                                    BLT     4$                  ;;BR IF DONE
054126  005202                                    INC     R2                  ;;INCREASE THE BCD DIGIT BY 1
054130  000774                                    BR      3$
054132  060105                           4$:      ADD     R1,R5               ;;ADD BACK THE CONSTANT
054134  005702                                    TST     R2                  ;;CHECK IF BCD DIGIT=0
054136  001002                                    BNE     5$                  ;;FALL THROUGH IF 0
054140  105716                                    TSTB    (SP)                ;;STILL DOING LEADING 0'S?
054142  100407                                    BMI     7$                  ;;BR IF YES
054144  106316                           5$:      ASLB    (SP)                ;;MSD?
054146  103003                                    BCC     6$                  ;;BR IF NO
054150  116663  000001  177777                    MOVB    1(SP),-1(R3)        ;;YES--SET THE SIGN
054156  052702  000060                   6$:      BIS     #'0,R2              ;;MAKE THE BCD DIGIT ASCII
054162  052702  000040                   7$:      BIS     #' ,R2              ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
054166  110223                                    MOVB    R2,(R3)+            ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
054170  005720                                    TST     (R0)+               ;;JUST INCREMENTING
054172  020027  000010                            CMP     R0,#10              ;;CHECK THE TABLE INDEX
054176  002746                                    BLT     2$                  ;;GO DO THE NEXT DIGIT
054200  003002                                    BGT     8$                  ;;GO TO EXIT
054202  010502                                    MOV     R5,R2               ;;GET THE LSD
054204  000764                                    BR      6$                  ;;GO CHANGE TO ASCII
054206  105726                           8$:      TSTB    (SP)+               ;;WAS THE LSD THE FIRST NON-ZERO?
054210  100003                                    BPL     9$                  ;;BR IF NO
054212  116663  177777  177776                    MOVB    -1(SP),-2(R3)       ;;YES--SET THE SIGN FOR TYPING
```

```
054220  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
054222  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
054224  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
054226  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
054230  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
054232  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
054234  104401  054262                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
054240  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
054246  012616                          MOV     (SP)+,(SP)
054250  000002                          RTI                     ;;RETURN TO USER
054252  023420                  $DTBL:  10000.
054254  001750                          1000.
054256  000144                          100.
054260  000012                          10.
054262  000004                  $DBLK:  .BLKW   4

                                .SBTTL  RANDOM NUMBER GENERATOR ROUTINE

                                ;;*************************************************************
                                ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
                                ;*WITH A RANGE OF 0 TO 2(+33)-1.
                                ;*CALL:
                                ;*      JSR     PC,$RAND        ;;CALL THE ROUTINE
                                ;*      RETURN                  ;;RETURN HERE THE RANDOM
                                ;*                              ;;NUMBER WILL BE IN
                                ;*                              ;;$HINUM,$LONUM

054272                          $RAND:
054272  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
054274  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
054276  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
054300  013700  054372                  MOV     $LONUM,R0       ;;SET R0 WITH LOW
054304  013701  054370                  MOV     $HINUM,R1       ;;SET R1 WITH HIGH
054310  012702  177771                  MOV     #-7,R2          ;;SET SHIFT COUNT
054314  006300                  1$:     ASL     R0              ;;SHIFT R0 LEFT AND
054316  006101                          ROL     R1              ;;ROTATE CARRY INTO R1 AND
054320  005202                          INC     R2              ;;CHECK FOR DONE
054322  001374                          BNE     1$              ;;CONTINUE SHIFT LOOP
054324  063700  054372                  ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
054330  005501                          ADC     R1              ;;PROPOGATE CARRY
054332  063701  054370                  ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
054336  062700  001057                  ADD     #1057,R0        ;;ADD LOW CONSTANT
054342  005501                          ADC     R1              ;;PROPOGATE CARRY
054344  062701  047401                  ADD     #47401,R1       ;;ADD HIGH CONSTANT
054350  010037  054372                  MOV     R0,$LONUM       ;;SAVE R0
054354  010137  054370                  MOV     R1,$HINUM       ;;SAVE R1
054360  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
054362  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
054364  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
054366  000207                          RTS     PC              ;;RETURN
054370  176543                  $HINUM: .WORD   176543
054372  123456                  $LONUM: .WORD   123456

                                .SBTTL  TRAP DECODER

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

```
                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                    ;*GO TO THAT ROUTINE.

054374  010046                      $TRAP:  MOV     R0,-(SP)         ;;SAVE R0
054376  016600  000002                      MOV     2(SP),R0         ;;GET TRAP ADDRESS
054402  005740                              TST     -(R0)            ;;BACKUP BY 2
054404  111000                              MOVB    (R0),R0          ;;GET RIGHT BYTE OF TRAP
054406  006300                              ASL     R0               ;;POSITION FOR INDEXING
054410  016000  054430                      MOV     $TRPAD(R0),R0    ;;INDEX TO TABLE
054414  000200                              RTS     R0               ;;GO TO ROUTINE


                                    ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

054416  011646                      $TRAP2: MOV     (SP),-(SP)       ;;MOVE THE PC DOWN
054420  016666  000004  000002              MOV     4(SP),2(SP)      ;;MOVE THE PSW DOWN
054426  000002                              RTI                      ;;RESTORE THE PSW

                                    .SBTTL  TRAP TABLE

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

                                    ;       ROUTINE
                                    ;       -------
054430  054416                      $TRPAD: .WORD   $TRAP2
054432  053336                              $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
054434  053644                              $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
054436  053620                              $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
054440  053660                              $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
054442  054046                              $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

054444  052536                              $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

054446  052466                              $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
054450  052750                              $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
054452  053070                              $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
054454  053242                              $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
054456  053300                              $RESREG ;;CALL=RESREG   TRAP+13(104413) RESTORE R0-R5 ROUTINE

054460  055574                              CLEAN   ;;CALL=RSET     TRAP+14(104414) GO RESET ALL REGISTERS.
054462  055544                              ABORTT  ;;CALL=SKIPT    TRAP+15(104415) THIS WILL SKIP TO THE NEXT TEST
054464  056256                              MMDES   ;;CALL=MMSKIP   TRAP+16(104416) IF SWITCH # IS ON SKIP TO THE NEXT TEST
054466  056300                              MSIZER  ;;CALL=SIZE     TRAP+17(104417) DETERMINE THE HIGHEST ADDRESS IN MEMORY
054470  055664                              SKBADR  ;;CALL=SKPBAD   TRAP+20(104420) SKIP TEST IF ERROR ADDRESS REGISTER IS I
054472  055710                              SKBERR  ;;CALL=SKPBER   TRAP+21(104421) SKIP TEST IF ERROR REGISTER IS INOPERATI
054474  055726                              SKBCNR  ;;CALL=SKPBCN   TRAP+22(104422) SKIP TEST IF CONTROL REGISTER IS INOPERA
054476  055744                              SKBMNR  ;;CALL=SKPBMN   TRAP+23(104423) SKIP TEST IF MAINTENANCE REGISTER IS INO
054500  055762                              SKBHMR  ;;CALL=SKPBHM   TRAP+24(104424) SKIP TEST IF HIT/MISS REGISTER IS IN OPE
054502  041752                              RANDWR  ;;CALL=WRRAND   TRAP+25(104425) FILL BUFFER WITH RANDOM SEQUENCE

054504  061046                              RS4HAN  ;;CALL=CALRS4   TRAP+26(104426) DO RS04 FUNCTION
054506  060076                              RP4HAN  ;;CALL=CALRP4   TRAP+27(104427) DO RP04 FUNCTION
054510  063542                              RH4HAN  ;;CALL=CALRH4   TRAP+30(104430) DO MBT FUNCTION
```

```
054512  062002                          RK5HAN  ;;CALL=CALRK5   TRAP+31(104431) DO RK05 FUNCTION
054514  063016                          UBEHAN  ;;CALL=CALUBE   TRAP+32(104432) DO UBE FUNCTION
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
054516  012737  054662  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
054524  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
054532  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
054534  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
054536  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
054540  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
054542  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
054544  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
054546  017746  124766                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
054552  010637  054666                  MOV     SP,$SAVR6       ;;SAVE SP
054556  012737  054570  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
054564  000000                          HALT
054566  000776                          BR      .-2             ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
054570  012737  054662  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
054576  013706  054666                  MOV     $SAVR6,SP       ;;GET SP
054602  005037  054666                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
054606  005237  054666          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
054612  001375                          BNE     1$              ;;OF  WORD
054614  012677  124720                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
054620  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
054622  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
054624  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
054626  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
054630  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
054632  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
054634  012737  054516  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
054642  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
054650  104401                          TYPE                    ;;REPORT THE POWER FAILURE
054652  066433                  $PWRMG: .WORD   POWERM          ;;POWER FAIL MESSAGE POINTER
054654  012716                          MOV     (PC)+,(SP)      ;;RESTART AT START
054656  004146                  $PWRAD: .WORD   START           ;;RESTART ADDRESS
054660  000002                          RTI
054662  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
054664  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
054666  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
                                .SBTTL  ROUTINE TO SIZE MEMORY

                                ;;*****************************************************************
                                ;*CALL:
                                ;*      JSR     PC,$SIZE
                                ;*      RETURN
                                ;*$LSTAD WILL CONTAIN:
                                ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
                                ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
                                ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
                                ;*$KT11 IS THE MEMORY MANAGEMENT KEY
                                ;*BIT07  0 DON'T USE MEMORY MANAGEMENT
```

```
                                      ;*      MUST BE SETUP BEFORE THE CALL
                                      ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
                                      ;*      DETERMINED BY ROUTINE
                                      ;* --NOTE--
                                      ;*THIS ROUTINE SUPPORTS PDP 11/74.
                                      ;*IF ACTUAL MEMORY IS LESS THAN THAT INDICATED BY THE SIZE REGISTER
                                      ;*AND A REFERENCE IS MADE TO A MEMORY ADDRESS THAT IS GREATER THAN
                                      ;*ACTUAL MEMORY BUT LESS THAN SIZE REGISTER (INDICATED), THEN A
                                      ;*MEMORY REFERENCE TIMEOUT TO VECTOR 114 WILL OCCUR.
                                      ;* --NOTE--
                                      ;*THIS ROUTINE WILL NOT SIZE FOR MEMORY GREATER THAN 1920K.

054670  010046                        $SIZE:  MOV     R0,-(SP)          ;;SAVE R0 ON THE STACK
054672  010146                                MOV     R1,-(SP)          ;;SAVE R1 ON THE STACK
054674  010246                                MOV     R2,-(SP)          ;;SAVE R2 ON THE STACK
054676  010346                                MOV     R3,-(SP)          ;;SAVE R3 ON THE STACK
054700  013746  000114                        MOV     @#114,-(SP)       ;;SAVE MEMORY ERROR VECTOR PS & PC        (REV F)
054704  013746  000116                        MOV     @#116,-(SP)       ;;                                       (REV F)
054710  012737  000116  000114                MOV     #116,@#114        ;;IGNORE PARITY ERRORS WHILE SIZING      (REV F)
054716  012737  000002  000116                MOV     #RTI,@#116        ;;                                       (REV F)
054724  013746  000004                        MOV     @#ERRVEC,-(SP)    ;;SAVE PRESENT ERROR VECTOR PS & PC
054730  013746  000006                        MOV     @#ERRVEC+2,-(SP)
054734  010600                                MOV     SP,R0             ;;SAVE THE STACK POINTER
                                      ;;SET THE ERRVEC PS TO THE PRESENT PS
054736  104400                                TRAP                      ;;PUSH OLD PSW AND PC ON STACK
054740  012637  000006                        MOV     (SP)+,@#ERRVEC+2      ;;SAVE THE PSW IN @#ERRVEC+2
054744  012701  003776                        MOV     #3776,R1          ;;SETUP ADDRESS
054750  105727                                TSTB    (PC)+             ;;USE MEMORY MANAGEMENT?
054752  000200                        $KT11:  .WORD   200               ;;SET TO USE MEMORY MANAGEMENT
054754  100065                                BPL     $CORE             ;;BR IF NO
054756  012737  055122  000004                MOV     #$KTNEX,@#ERRVEC  ;;SET FOR TIMEOUT
054764  005737  177572                        TST     @#SR0             ;;KT11 ARE YOU THERE?
054770  052737  100000  054752                BIS     #100000,$KT11     ;;YES--SET KT11 KEY
054776  005046                                CLR     -(SP)             ;;INITIALIZE FOR 'PAR'' LOADING
055000  012702  172340                        MOV     #KIPAR0,R2        ;;ADDRESS OF FIRST 'PAR''
055004  012703  000010                        MOV     #^D8,R3           ;;LOAD EIGHT 'PAR.'S'' AND EIGHT 'PDR.'S''
055010  012762  077406  177740        1$:     MOV     #77406,-40(R2)    ;;PDR = 4K, UP, READ/WRITE
055016  011622                                MOV     (SP),(R2)+        ;;LOAD 'PAR''
055020  062716  000200                        ADD     #200,(SP)         ;;UPDATE FOR NEXT 'PAR''
055024  077307                                SOB     R3,1$             ;;LOOP UNTIL ALL EIGHT ARE LOADED
055026  012742  177600                        MOV     #177600,-(R2)     ;;SETUP KIPAR7 FOR I/O
055032  005042                                CLR     -(R2)             ;;SETUP KIPAR6 FOR TESTING
055034  012737  055052  000004                MOV     #2$,@#ERRVEC      ;;CATCH TIMEOUT IF NO SR3
055042  012737  000020  172516                MOV     #20,@#SR3         ;;ENABLE 22 BIT MODE
055050  000401                                BR      3$                ;;THIS PDP-11 HAS A SR3 REGISTER
055052  022626                        2$:     CMP     (SP)+,(SP)+       ;;CLEAN OFF THE STACK--NO SR3
055054  005237  177572                3$:     INC     @#SR0             ;;TURN ON MEMORY MANAGEMENT
055060  012737  055112  000004                MOV     #$KTOUT,@#ERRVEC  ;;SET FOR TIME OUT
055066  012737  055234  000114                MOV     #$MTMOUT,@#114    ;;SET FOR MEM REF TIMEOUT
055074  005737  143776                4$:     TST     @#143776          ;;TRAP ON NON-EX-MEM
055100  062712  000040                        ADD     #40,(R2)          ;;MAKE A 1K STEP
055104  022712  170000                        CMP     #170000,(R2)      ;;LAST ONE?
055110  101371                                BHI     4$                ;;NO--TRY IT
055112  011202                        $KTOUT: MOV     (R2),R2           ;;GET LAST BANK+1
055114  005037  177572                        CLR     @#SR0             ;;TURN OFF MEMORY MANAGEMENT
055120  000421                                BR      $SIZEX
```

```
055122  042737  100000  054752  $KTNEX: BIC     #100000,$KT11       ;;KT11 NON-EXISTENT
055130  012737  055160  000004  $CORE:  MOV     #$CROUT,@#ERRVEC    ;;SET FOR TIMEOUT
055136  005002                          CLR     R2                  ;;SET UP BANK
055140  062701  004000          1$:     ADD     #4000,R1            ;;INCREMENT BY 1K
055144  062702  000040                  ADD     #40,R2              ;;1K STEP
055150  005711                          TST     (R1)                ;;TRAP ON TIME OUT
055152  022701  177776                  CMP     #177776,R1          ;;LAST ONE
055156  001370                          BNE     1$                  ;;NO--TRY AGAIN
055160  162701  004000          $CROUT: SUB     #4000,R1
055164  162702  000040          $SIZEX: SUB     #40,R2              ;;DROP BACK
055170  010006                          MOV     R0,SP               ;;RESTORE THE STACK
055172  012637  000006                  MOV     (SP)+,@#ERRVEC+2    ;;RESTORE ERROR VECTOR
055176  012637  000004                  MOV     (SP)+,@#ERRVEC
055202  012637  000116                  MOV     (SP)+,@#116         ;;RESTORE MEMORY ERROR VECTOR               (REV F)
055206  012637  000114                  MOV     (SP)+,@#114         ;;                                          (REV F)
055212  010137  055266                  MOV     R1,$LSTAD           ;;LAST ADDRESS
055216  010237  055270                  MOV     R2,$LSTBK           ;;LAST BANK
055222  012603                          MOV     (SP)+,R3            ;;RESTORE R3
055224  012602                          MOV     (SP)+,R2            ;;RESTORE R2
055226  012601                          MOV     (SP)+,R1            ;;RESTORE R1
055230  012600                          MOV     (SP)+,R0            ;;RESTORE R0
055232  000207                          RTS     PC
055234  032737  000001  177744  $MTMOUT: BIT    #BIT0,@#MEMERR      ;;MAKE SURE TRAP TO 114 IS DUE
055242  001005                          BNE     1$                  ;;TO MEMORY REFERENCE TIMEOUT
                                                                    ;;IF NOT, IS IT AN ABORT?
055244  032737  100000  177744          BIT     #BIT15,@#MEMERR     ;;CPU ABORT?
055252  001001                          BNE     1$                  ;;IF YES, EXIT OUT
055254  000002                          RTI                         ;;IF NOT, CONTINUE
055256  012737  177777  177744  1$:     MOV     #-1,@#MEMERR        ;;CLEAR THE MEM ERROR REG
055264  000712                          BR      $KTOUT
055266  000000                  $LSTAD: .WORD   0                   ;;CONTAINS THE LAST ADDRESS
055270  000000                  $LSTBK: .WORD   0                   ;;CONTAINS THE LAST BANK

                                .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

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


055272  104412                  $DB20:  SAVREG                      ;;SAVE ALL REGISTERS
055274  016601  000002                  MOV     2(SP),R1            ;;PICKUP THE POINTER TO LOW WORD
055300  012705  055411                  MOV     #$OCTVL+13.,R5      ;;POINTER TO DATA TABLE
055304  012704  000014                  MOV     #12.,R4             ;;DO ELEVEN CHARACTERS
055310  012703  177770                  MOV     #^C7,R3             ;;MASK
055314  012100                          MOV     (R1)+,R0            ;;LOWER WORD
055316  012101                          MOV     (R1)+,R1            ;;HIGH WORD
055320  005002                          CLR     R2                  ;;TERMINATOR
055322  110245                  1$.     MOVB    R2,-(R5)            ;;PUT CHARACTER IN DATA TABLE
055324  010002                          MOV     R0,R2               ;;GET THIS DIGIT
055326  005304                          DEC     R4                  ;;COUNT THIS CHARACTER
055330  003007                          BGT     3$                  ;;BR IF NOT THE LAST DIGIT
```

```
055332  001405                              BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
055334  005205                              INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
055336  010566  000002                      MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
055342  104413                              RESREG                  ;;RESTORE ALL REGISTERS
055344  000207                              RTS     PC              ;;RETURN TO USER
055346  006203                      2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
055350  006001                      3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
055352  006000                              ROR     R0              ;;      THE NEXT OCTAL DIGIT
055354  006001                              ROR     R1
055356  006000                              ROR     R0
055360  006001                              ROR     R1
055362  006000                              ROR     R0
055364  040302                              BIC     R3,R2           ;;MASK OUT ALL JUNK
055366  062702  000060                      ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
055372  000753                              BR      1$              ;;GO PUT IT IN THE DATA TABLE
055374  000016                      $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE

                                    ;THIS ROUTINE IS CALLED BY UNEXPECTED TRAPS TO VECTOR ERRVEC.
                                    ;THE ERROR IS REPORTED AND CONTROL IS TRANSFERRED BACK TO THE TEST
                                    ;FOLLOWING THE ONE THAT WAS INTERRUPTED WHEN THE ERROR OCCURRED.
055412  011637  001634              CPSPUR: MOV     (SP),$TMP1
055416  012737  055434  001636              MOV     #1$,$TMP2
055424  013737  177766  001640              MOV     @#CPUERR,$TMP3
055432  022626                              CMP     (SP)+,(SP)+     ;RESET THE STACK
055434  104150                      1$:     ERROR   150
055436  104415                              SKIPT

                                    ;THIS ROUTINE HANDLE UNEXPECTED TRAPS TO #CACHVEC.
055440  012737  055536  000114      SPUR:   MOV     #10$,@#CACHVEC
055446  013700  177744                      MOV     @#MEMERR,R0
055452  032700  000014                      BIT     #14,R0          ;SEE IF IT WAS A MAIN MEMORY ERROR.
055456  001405                              BEQ     9$
055460  013701  177740                      MOV     @#LOADRS,R1     ;IF SO THERE IS BAD PARITY IN THE
055464  042701  176000                      BIC     #176000,R1      ;CACHE AND IT MUST BE PURGED!!.
055470  005711                              TST     (R1)
055472  012737  055440  000114      9$:     MOV     #SPUR,@#CACHVEC
055500  013737  177744  001642              MOV     @#MEMERR,$TMP4  ;TRAP HERE IF AN UNEXPECTED
055506  013737  177740  001634              MOV     @#LOADRS,$TMP1  ;ERROR, PARITY, OCCURS.
055514  013737  177742  001636              MOV     @#HIADRS,$TMP2
055522  011637  001640                      MOV     (SP),$TMP3
055526  022626                              CMP     (SP)+,(SP)+
055530  104014                      1$:     ERROR   14
055532  000005                              RESET                   ;TO STOP THE ACTION OF ANY I/O DEVICE   .
055534  104415                              SKIPT                   ;?????
055536  022626                      10$:    CMP     (SP)+,(SP)+
055540  000137  055472                      JMP     9$

                                    ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL SKIPT.
                                    ;IT TELLS THE USER THAT THE CURRENT TEST HAS BEEN
                                    ;ABORTED AND THAT CONTROL IS BEING PASSED TO THE NEXT TEST.
055544  011637  001634              ABORTT: MOV     (SP),$TMP1
055550  112737  000015  001514              MOVB    #15,$ITEMB
055556  022626                              CMP     (SP)+,(SP)+
055560  004737  056354                      JSR     PC,ERTYPE
055564  104414                              RSET
055566  000177  000000                      JMP     @SKAD           ;GO TO @SKAD, WHICH SHOULD
```

```
                                                                ;BE SET TO THE
055572  000000                          SKAD:   .WORD   0       ;ADDRESS OF THE NEXT TEST.


                                        ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL RSET. IT CLEARS ALL
                                        ;THE IMPORTANE REGISTERS AND RESETS THE STACK.
055574                                  CLEAN:

055574  012737  055440  000114                  MOV     #SPUR,@#CACHVEC
055602  012737  055412  000004                  MOV     #CPSPUR,@#ERRVEC
055610  011637  055662                          MOV     (SP),BACKAD
055614  012706  001400                          MOV     #STACK,SP
055620  005037  177750                          CLR     @#MAINT         ;CLEAR ALL CONTROL AND ERROR
055624  005037  177572                          CLR     @#MMR0          ;REGISTERS.
055630  005037  172516                          CLR     @#MMR3
055634  005037  177746                          CLR     @#CONTRL
055640  012737  177777  177744                  MOV     #-1,@#MEMERR
055646  005037  177766                          CLR     @#CPUERR
055652  005037  177776                          CLR     @#PSW
055656  000177  000000                          JMP     @BACKAD
055662  000000                          BACKAD: .WORD   0


                                        ;COME HERE TO TEST THE REGISTER FLAGS AND USE THEM TO DETERMINE WHETHER
                                        ;OR NOT TO SKIP A TEST WHICH RELIES ON THE FUNCTIONALLITY OF THAT REGISTER
                                        ;TO BE PROPERLY RUN.
                                        ;THESE ROUTINES ARE CALLED BY THE TRAP CATCHER CALLS:
                                        ;       SKPBAD          SKIPT IF BAD ERROR ADDRESS REGISTER
                                        ;       SKPBER          SKIPT IF BAD ERROR REGISTER
                                        ;       SKPBCN          SKIPT IF BAD CONTROL REGISTER
                                        ;       SKPBMN          SKIPT IF BAD MAINTENANCF REGISTER
                                        ;       SKPBHM          SKIPT IF BAD HIT/MISS REGISTER
                                        ;

055664  005737  056002                  SKBADR: TST     LOAFLG
055670  001004                                  BNE     1$
055672  005737  056004                          TST     HIAFLG
055676  001001                                  BNE     1$
055700  000002                                  RTI
055702  104401                          1$:     TYPE
055704  067415                                  .WORD   ADRNG
055706  000433                                  BR      SKRNG

055710  005737  056006                  SKBERR: TST     MMRFLG
055714  001001                                  BNE     1$
055716  000002                                  RTI
055720  104401                          1$:     TYPE
055722  067525                                  .WORD   ERRNG
055724  000424                                  BR      SKRNG

055726  005737  056010                  SKBCNR: TST     CONFLG
055732  001001                                  BNE     1$
055734  000002                                  RTI
055736  104401                          1$:     TYPE
055740  067625                                  .WORD   CNRNG
055742  000415                                  BR      SKRNG
```

```

055744  005737  056012          SKBMNR: TST     MANFLG
055750  001001                          BNE     1$
055752  000002                          RTI
055754  104401                  1$:     TYPE
055756  067727                          .WORD   MNRNG
055760  000406                          BR      SKRNG

055762  005737  056014          SKBHMR: TST     HIMFLG
055766  001001                          BNE     1$
055770  000002                          RTI
055772  104401                  1$:     TYPE
055774  070035                          .WORD   HMRNG

055776  022626                  SKRNG:  CMP     (SP)+,(SP)+             ;RESET THE STACK AND GO TO THE
056000  104415                          SKIPT                           ;NEXT TEST!!!!.

056002  000000                  LOAFLG: .WORD   0                       ;THESE ARE FLAGS USED TO DESIGNATE
056004  000000                  HIAFLG: .WORD   0                       ;EITHER A GOOD OR A BAD REGISTER.
056006  000000                  MMRFLG: .WORD   0                       ;GOOD WILL BE DESIGNATED BY A
056010  000000                  CONFLG: .WORD   0                       ;0 BAD BY A NOT ZERO!!
056012  000000                  MANFLG: .WORD   0
056014  000000                  HIMFLG: .WORD   0
056016  000000                  LOAFL2: .WORD   0
056020  000000                  HIAFL2: .WORD   0
056022  000000                  MMRFL2: .WORD   0
056024  000000                  CONFL2: .WORD   0
056026  000000                  MANFL2: .WORD   0
056030  000000                  HIMFL2: .WORD   0

                                ;THIS ROUTINE IS CALLED TO DETERMINE THE PARITY OF
                                ;A DATA PATTERN. THE PATTERN WHICH IS TAKEN BY THIS
                                ;ROUTINE AS ITS ARGUMENT SHOULD BE PUT IN R0. THEN
                                ;TRANSFER CONTROL HERE BY EXECUTING:
                                ;       JSR     PC,PARCNT
                                ;WHEN THIS ROUTINE RETURNS THE NUMBER OF ON,(1), BITS
                                ;IN R0 IS LEFT IN R2. THIS WOULD BE A NUMBER BETWEEN
                                ;0 AND 16.
056032  012701  000001          PARCNT: MOV     #1,R1
056036  005002                          CLR     R2
056040  030100                  1$:     BIT     R1,R0
056042  001401                          BEQ     2$
056044  005202                          INC     R2
056046  006301                  2$:     ASL     R1
056050  103373                          BCC     1$
056052  000207                          RTS     PC

                                ;THIS ROUTINE IS CALLED TO RESTORE THE TOP 1500 (DEC) WORDS IN THE
                                ;FIRST 28K OF MEMORY. THIS SHOULD EFFECTIVELY RESTORE ANY MONITOR
                                ;OR LOADER THAT WAS PRESENT BEFORE THIS PROGRAM BEGAN EXECUTION.
                                ;CONTROL IS PASSED TO THIS ROUTINE BY AN INTERRUPT FROM THE TTY KEYBOARD
                                ;WHEN ANY CHARACTER IS TYPED ON THE KEYBOARD. IF THE CHARACTER
                                ;IS A ^C MEMORY IS RESTORED. IF IT'S A ^G THE PROGRAM REQUESTS
                                ;A NEW SOFTWARE SWR.
                                ; A RETURN IS MADE TO THE TEST FOLLOWING
                                ;THE ONE WHOSE EXECUTION WAS INTERRUPTED BY THE KEYBOARD INTERRUPT.
```

```
056054 005037 177750         RESMON: CLR     @#MAINT
056060 017700 123462                 MOV     @$TKB,R0
056064 104414                        RSET
056066 005003                        CLR     R3
056070 042700 177600                 BIC     #177600,R0        ;GET THE CHARACTER
056074 022700 000003                 CMP     #3,R0             ;SEE IF IT'S ^C
056100 001037                        BNE     NOCNC             ;BRANCH AND GO TO NEXT TEST IF NOT.
056102 104401                        TYPE                      ;ECHO THE CONTROL-C AS '^C'
056104 066370                        .WORD   CONCMS
056106 012704 002734         CHAINQ: MOV     #^D1500,R4                ;AND RESTORE THE MONITOR.
056112 012701 120314                 MOV     #BOTTOM+4,R1
056116 012702 160000                 MOV     #160000,R2
056122 012142                1$:     MOV     (R1)+,-(R2)
056124 077402                        SOB     R4,1$
056126 012737 177777 056254          MOV     #-1,MONF          ;RESET THE MONITOR RESTORED FLAG.
056134 022703 125252                 CMP     #125252,R3
056140 001001                        BNE     STOP
056142 000207                        RTS     PC                ;IF THE MONITOR WAS RESTORED BY THE
                                                               ;.$EOP ROUTIN RETURN TO .$EOP.
                                                               ;OTHERWISE HALT.
056144 005737 000042         STOP:   TST     @#42              ;IN AUTO MODE?
056150 001402                        BEQ     1$                ;BRANCH IF NOT
056152 000137 051454                 JMP     $GET42+16         ;ELSE RETURN TO MONITOR
056156 104401                1$:     TYPE                      ;TYPE THE MONITOR RESTORED MESSAGE.
056160 066374                        .WORD   MMESRS
056162 013737 056252 000060          MOV     MONTTY,@#TKVEC    ;SET THE TTY KEYBOARD INTERRUPT VECTOR
                                                               ;TO ITS INITIAL VALUE.
056170 000000                        HALT                      ;AND HALT!!
056172 012737 056054 000060          MOV     #RESMON,@#TKVEC
056200 022737 000176 001540  NOCNC:  CMP     #SWREG,SWR        ;SOFTWARE SWR SELECTED?
056206 001012                        BNE     1$                ;BRANCH IF NOT
056210 022700 000007                 CMP     #7,R0             ;IS IT ^G?
056214 001007                        BNE     1$                ;BRANCH IF NOT
056216 123727 001534 000001          CMPB    $AUTOB,#1         ;ARE WE RUNNING IN AUTO MODE?
056224 001403                        BEQ     1$                ;BRANCH IF  YES
056226 104401 053213                 TYPE    ,$CNTLG           ;ECHO ^G
056232 104406                        GTSWR                     ;GET NEW SWR SETTING
056234 005077 123306         1$:     CLR     @$TKB             ;NOT CONTROL C SO RETURN TO NEXT TEST.
056240 152777 000100 123276          BISB    #BIT6,@$TKS
056246 000177 177320                 JMP     @SKAD             ;RETURN.
056252 000000                MONTTY: .WORD   0        ;STORAGE FOR THE TTY KEYBOARD VECTOR'S ORIGINAL
                                                               ;CONTENTS.
056254 177777                MONF:   .WORD   177777            ;FLAG. IF NOT -1 THE MONITOR IS SAVED.


                             ;THIS ROUTINE IS CALLED BY THE TRAP CALL MMSKIP. IT LOOKS
                             ;AT THE SWITCH REGISTER AND DETERMINES WHETHER OR NOT
                             ;SWITCH #7 IS ON. IF SO THE CURRENT TEST IS SKIPPED
                             ;AND THE NEXT TEST IS ENTERED. A SSKAD MUST BE ISSUED
                             ;BEFORE THE MMSKIP.
                             ;THE PURPOSE OF SWITCH #7 IS TO CAUSE THE DELETION OF THE
                             ;EXECUTION OF ANY TEST WHICH RELIES ON MEMORY MANAGEMENT
                             ;FOR ITS OPERATION.

056256 032777 000200 123254  MMDES:  BIT     #SW7,@SWR
056264 001001                        BNE     1$                ;IS THE SWITCH ON?
```

```
056266  000002                          RTI                             ;NO. SO RETURN.
056270  022626                  1$:     CMP     (SP)+,(SP)+
056272  104414                          RSET
056274  000177  177272                  JMP     @SKAD                   ;YES. GO TO THE NEXT TEST.
                                ;THIS ROUTINE IS CALLED TO DETERMINE THE HIGHEST POSSIBLE
                                ;ADDRESS IN MEMORY. IT IS CALLED THUS, BY TRAP CALL SIZE:
                                ;               SIZE
                                ;       LOORDA: .WORD   0
                                ;       HIORDA: .WORD   0
                                ;       NXTINST:
                                ;THE LOW ORDER 16-BITS OF THE ADDRESS ARE LEFT IN THE
                                ;WORD DIRECTLY FOLLOWING THE CALL. THE HIGH ORDER 6-BITS
                                ;ARE LEFT IN THE NEXT WORD AND CONTROL IS RETURNED
                                ;TO THE THIRD WORD FOLLOWING THE CALL.
056300  010046                  MSIZER: MOV     R0,-(SP)                ;SAVE THE CONTENTS OF R0 AND R1
056302  010146                          MOV     R1,-(SP)                ;GET THE ADDRESS OF
056304  016600  000004                  MOV     4(SP),R0                ;THE CALL OF THE STACK.
056310  013710  055270                  MOV     $LSTBK,(R0)             ;GET THE ACCESSABLE BOUNDARY OF MEMORY
056314  005060  000002                  CLR     2(R0)
056320  012701  000006                  MOV     #6,R1                   ;ROTATE THE 16-BIT 'BLOCK'
                                                                        ;NUMBER 6-BITS TO THE
056324  006310                  1$:     ASL     (R0)                    ;LEFT AND TURN ON LOW ORDER
056326  006160  000002                  ROL     2(R0)                   ;BITS 1-5 LEAVING BIT-0
056332  077104                          SOB     R1,1$                   ;OFF SO AS TO CREATE
056334  052710  000076                  BIS     #76,(R0)                ;THE 22-BIT PHYSICAL ADDRESS OF
                                                                        ;THE HIGHEST WORD IN
                                                                        ;MEMORY.
056340  022020                          CMP     (R0)+,(R0)+             ;DETERMINE THE RETURN ADDRESS
056342  010066  000004                  MOV     R0,4(SP)                ;AND LEAVE ON THE STACK FOR
                                                                        ;AN RTI.
056346  012601                          MOV     (SP)+,R1                ;RESTORE R1 AND R0.
056350  012600                          MOV     (SP)+,R0
056352  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
056354  104401                  ERTYPE: TYPE
056356  001713                          .WORD   $CRLF
056360  010046                          MOV     R0,-(SP)                ;SAVE R0
056362  005000                          CLR     R0
056364  113700  001514                  MOVB    $ITEMB,R0               ;GET THE ITEM NUMBER
056370  001005                          BNE     1$                      ;ZERO?
056372  013746  001516                  MOV     $ERRPC,-(SP)            ;YES, TYPE JUST THE PC
056376  104402                          TYPOC                           ;OF THE ERROR CALL.
056400  000137  056716                  JMP     ERT5

056404  005300                  1$:     DEC     R0                      ;MAKE R0 AN INDEX FOR THE
056406  072027  000003                  ASH     #3,R0                   ;ERROR TABLE
056412  062700  001752                  ADD     #$ERRTB,R0
056416  012037  056426                  MOV     (R0)+,2$                ;TYPE EM, ERROR MESSAGE.
056422  001404                          BEQ     3$
056424  104401                          TYPE
056426  000000                  2$:     .WORD   0
056430  104401                          TYPE
```

```
056432  001713                          .WORD   $CRLF
056434  012037  056444          3$:     MOV     (R0)+,4$        ;TYPE DH, DATA HEADER
056440  001404                          BEQ     5$
056442  104401                          TYPE
056444  000000                  4$:     .WORD   0
056446  104401                          TYPE
056450  001713                          .WORD   $CRLF
056452  010146                  5$:     MOV     R1,-(SP)        ;SAVE R1
056454  012001                          MOV     (R0)+,R1        ;GET DT, DATA TABLE ADDRESS
056456  001002                          BNE     6$
056460  000137  056714                  JMP     ERT4            ;JMP IF NO ERROR TABLE.
056464  012000                  6$:     MOV     (R0)+,R0        ;GET DF, DATA FORMAT ADDRESS
056466  105710                  ERT1:   TSTB    (R0)            ;DATA FORMAT ENTRY EQUALS
056470  001003                          BNE     7$              ;ZERO?
056472  013146                          MOV     @(R1)+,-(SP)    ;YES, SO TYPE A 16-BIT
056474  104402                          TYPOC                   ;OCTAL NUMBER
056476  000500                          BR      ERT2
056500  122710  000001          7$:     CMPB    #1,(R0)         ;FORMAT EQUALS 1?
056504  001003                          BNE     8$
056506  013146                          MOV     @(R1)+,-(SP)    ;YES, TYPE A DECIMAL NUMBER
056510  104405                          TYPDS
056512  000472                          BR      ERT2

056514  122710  000002          8$:     CMPB    #2,(R0)         ;FORMAT 2?
056520  001012                          BNE     9$
056522  012146                  85$:    MOV     (R1)+,-(SP)     ;YES, TYPE A 22-BIT NUMBR
056524  004737  055272                  JSR     PC,$DB20        ;CALL $DB20 TO CONVERT THE
056530  062716  000003                  ADD     #3,(SP)         ;BINARY TO ASCII
056534  012637  056542                  MOV     (SP)+,29$       ;TYPE THE STRING
056540  104401                          TYPE
056542  000000                  29$:    .WORD   0
056544  000455                          BR      ERT2

056546  122710  000004          9$:     CMPB    #4,(R0)         ;FORMAT 4?
056552  001004                          BNE     10$
056554  013146                          MOV     @(R1)+,-(SP)    ;YES, TYPE A 16-BIT
056556  104403                          TYPOS                   ;OCTAL NUMBER SUPRESSING
056560     016                          .BYTE   16              ;LEADING ZEROES
056561     000                          .BYTE   0
056562  000446                          BR      ERT2
056564  122710  000003          10$:    CMPB    #3,(R0)         ;FORMAT 3?
056570  001007                          BNE     11$
056572  013146                          MOV     @(R1)+,-(SP)    ;YES CONVERT 16-BIT
056574  012737  177777  056722          MOV     #-1,TVADFL      ;VIRTUAL ADDRESS TO 32-BIT
056602  004737  056730                  JSR     PC,TYPVAD       ;PHYSICAL ADDRESS AND TYPE
056606  000434                          BR      ERT2            ;RELOCATE ONLY IF SEG. IS ON!
056610  122710  000005          11$:    CMPB    #5,(R0)         ;FORMAT 5?
056614  001005                          BNE     12$
056616  012137  056624                  MOV     (R1)+,20$       ;PRINT ASCIZ STRING
056622  104401                          TYPE
056624  000000                  20$:    .WORD   0
056626  000426                          BR      ERT3

056630  122710  000006          12$:    CMPB    #6,(R0)         ;FORMAT 6
056634  001005                          BNE     13$
056636  005037  056722                  CLR     TVADFL
```

```
056642  004737  056730                      JSR     PC,TYPVAD
056646  000414                              BR      ERT2

056650  122710  000007            13$:      CMPB    #7,(R0)         ;FORMAT 7?
056654  001010                              BNE     14$

056656  012146                              MOV     (R1)+,-(SP)
056660  004737  055272                      JSR     PC,$DB20
056664  012637  056672                      MOV     (SP)+,45$
056670  104401                              TYPE
056672  000000                    45$:      .WORD   0
056674  000401                              BR      ERT2

056676  000000                    14$:      HALT                    ;?????

056700  104401                    ERT2:     TYPE                    ;PRINT A TAB AFTER TYPING AN
056702  066500                              .WORD   $TAB            ;ERROR TABLE ENTRY OF ALL MODES
                                                                    ;EXCEPT ASCIZ
056704  005200                    ERT3:     INC     R0              ;POINT TO THE NEXT FORMAT BYTE
056706  005711                              TST     (R1)            ;IS THERE ANOTHER ENTRY?
056710  001401                              BEQ     ERT4
056712  000665                              BR      ERT1            ;YES, PROCESS IT
                                                                    ;OTHERWISE:
056714  012601                    ERT4:     MOV     (SP)+,R1        ;RESTORE R1
056716  012600                    ERT5:     MOV     (SP)+,R0        ;RESTORE R0
056720  000207                              RTS     PC              ;AND RETURN

056722  000000                    TVADFL: .WORD     0               ;FLAG USED TO TELL TYVAD
                                                                    ;WHETHER TO CONDITIONALLY
                                                                    ;OR UNCONDITIONALLY RELOCATE
                                                                    ;WHEN TYPING AN ADDRESS,
                                                                    ;-1 OR 0 RESPECTIVELY

056724  000000                    TVADLO: .WORD     0               ;REGISTERS FOR THE 22-BIT
056726  000000                    TVADHI: .WORD     0               ;ADDRESS COMPUTED BY TYVAD.

                                  ;ROUTINE WHICH CONVERTS A 16-BIT ADDRESS TO A 22-BIT
                                  ;ADDRESS. IF TVADFL IS -1, THEN CONVERT TO THE 22-BIT
                                  ;REAL ADDRESS DEPENDENT ON SEG BEING ON OR OFF FOR RELOCATION.
                                  ;IF TVADFL IS ZERO THEN UNCONDITIONAL USE THE KERNAL
                                  ;PAR WHICH IS APPROPIATE TO DO RELOCATION.
056730  104412                    TYPVAD: SAVREG
056732  016601  000002                      MOV     2(SP),R1        ;GET THE VIRTUAL
056736  010137  056724                      MOV     R1,TVADLO       ;ADDRESS
056742  005037  056726                      CLR     TVADHI
056746  005737  056722                      TST     TVADFL          ;CONDITIONALLY RELOCATE?
056752  001404                              BEQ     1$
056754  032737  000001  177572              BIT     #1,@#MMR0       ;YES, SEE IF MEMORY
056762  001424                              BEQ     2$              ;MANAGEMENT IS ON
056764  005000                    1$:       CLR     R0              ;RELOCATE
056766  073027  000003                      ASHC    #3,R0           ;LEFT SHIFT R0 AND R1
056772  006300                              ASL     R0              ;THREE PLACES. R0 ONE
                                                                    ;MORE SO THAT IT CONTAINS
                                                                    ;2 X THE UPPER 3-BITS OF
056774  000241                              CLC                     ;THE VIRTUAL ADDRESS
056776  006001                              ROR     R1              ;RESTORE R1 TO THE OFFSET
```

```
057000  006001                                  ROR     R1                      ;OF THE VIRTUAL ADDRESS
057002  006001                                  ROR     R1                      ;TO THE PAR
057004  062700  172340                          ADD     #KIPAR0,R0              ;DETERMINE THE CORRECT PAR'S
                                                                                ;ADDRESS
057010  011003                                  MOV     (R0),R3                 ;GET ITS CONTENTS
057012  005002                                  CLR     R2
0570 4  073227  000006                          ASHC    #6,R2                   ;MAKE THE BLOCK COUNT
                                                                                ;A 22-BIT ADDRESS.
05702C  060103                                  ADD     R1,R3                   ;ADD THE OFFSET TO THE
057022  005502                                  ADC     R2                      ;BASE ADDRESS

057024  010237  056726                          MOV     R2,TVADHI
057030  010337  056724                          MOV     R3,TVADLO
057034  012746  056724                  2$:     MOV     #TVADLO,-(SP)           ;CALL $DB20 TO .ONVERT THE
057040  004737  055272                          JSR     PC,$DB20                ;22-BIT
057044  062716  000003                          ADD     #3,(SP)                 ;TYPE ONLY 8 DIGITS.
057050  012637  057056                          MOV     (SP)+,3$
057054  104401                                  TYPE
057056  000000                          3$:     .WORD   0
057060  104413                                  RESREG                          ;RESTORE THE REGISTERS
057062  012616                                  MOV     (SP)+,(SP)              ;LEAVE ONLY THE RETURN
                                                                                ;ADDRESS ON THE STACK.
057064  000207                                  RTS     PC                      ;RETURN

                                                .SBTTL  SYSTEM DEVICE SIZER


                                        ;THIS ROUTINE .S CALLED TO DETERMINE WHAT
                                        ;CONTROLLERS AND WHAT DRIVES ARE AVAILABLE ON
                                        ;THE SYSTEM.
                                        ;IT USES THE FLAGS:
                                        ;                       RS4DFL
                                        ;                       RP4DFL
                                        ;                       RH4DFL
                                        ;                       RK5DFL
                                        ;                       UBEDFL
                                        ;WHICH ARE BYTES CONTAINING A BIT FOR EACH
                                        ;POSSIBLE DEVICE ON THE CONTROLLER
057066  005037  061002                  SIZDEV: CLR     RS4FLG                  ;INITIALIZE FLAGE
057072  005037  060032                          CLR     RP4FLG
057076  005037  063476                          CLR     RH4FLG
057102  005037  061736                          CLR     RK5FLG
057106  005037  062752                          CLR     UBEFLG
057112  005037  061004                          CLR     RS4ER1
057116  005037  060034                          CLR     RP4ER1
057122  005037  063500                          CLR     RH4ER1
057126  005037  061740                          CLR     RK5ER1
057132  005037  062754                          CLR     UBEER1
057136  104412                                  SAVREG
057140  105037  057460                          CLRB    RS4DFL
057144  105037  057461                          CLRB    RP4DFL
057150  105037  057462                          CLRB    RH4DFL
057154  105037  057463                          CLRB    RK5DFL
057160  105037  057464                          CLRB    UBEDFL

057164  013737  000004  057466                  MOV     @#4,SIZTM1              ;SAVE 4
057172  012737  057220  000004                  MOV     #1$,@#4                 ;IN CASE NON-EXISTENT REG.
```

```
10653
10654  057200  005777  124520                   TST     @RS4CS1         ;TEST FOR RS04
10655  057204  004737  057500                   JSR     PC,SETREG       ;GET THE REG ADD
10656  057210  003722                           .WORD   RS4REG
10657  057212  004737  057540                   JSR     PC,SIZRS4       ;GET THE # OF DRIVES
10658  057216  000403                           BR      2$
10659
10660  057220  022626                   1$:     CMP     (SP)+,(SP)+     ;THERE WAS NO RS04
10661  057222  005037  177766                   CLR     @#CPUERR
10662
10663  057226  012737  057254  000004   2$:     MOV     #3$,@#4         ;IN CASE NON-EXISTENT REG.
10664
10665  057234  005777  124522                   TST     @RP4CS1         ;TEST FOR RP04
10666  057240  004737  057500                   JSR     PC,SETREG       ;GET THE REG ADD
10667  057244  003760                           .WORD   RP4REG
10668  057246  004737  057624                   JSR     PC,SIZRP4       ;GET THE # OF DRIVES
10669  057252  000403                           BR      6$
10670
10671  057254  022626                   3$:     CMP     (SP)+,(SP)+     ;THERE WAS NO RP04
10672  057256  005037  177766                   CLR     @#CPUERR
10673
10674  057262  012737  057350  000004   6$:     MOV     #7$,@#4         ;IN CASE NON-EXISTENT REG.
10675  057270  005777  124544                   TST     @RH4CS1         ;TEST FOR MASS BUS TESTER
10676  057274  004737  057500                   JSR     PC,SETREG       ;GET THE REG ADD
10677  057300  004036                           .WORD   RH4REG
10678  057302  012777  000007  124540           MOV     #7,@RH4CS2      ;SET THE DRIVE #
10679  057310  022777  000040  124550           CMP     #40,@RH4DT
10680  057316  001017                           BNE     8$
10681  057320  013737  004040  004072           MOV     RH4CS1,RH4AE
10682  057326  062737  000074  004072           ADD     #74,RH4AE
10683  057334  004737  057500                   JSR     PC,SETREG
10684  057340  004070                           .WORD   RH4REX
10685
10686  057342  004737  060022                   JSR     PC,SIZRH4
10687  057346  000403                           BR      8$
10688
10689  057350  022626                   7$:     CMP     (SP)+,(SP)+     ;THEREE WAS NO MBT
10690  057352  005037  177766                   CLR     @#CPUERR
10691
10692  057356  012737  057404  000004   8$:     MOV     #9$,@#4         ;IN CASE NON-EXISTENT REG.
10693  057364  005777  124510                   TST     @RK5DS          ;TEST FOR RK05
10694  057370  004737  057500                   JSR     PC,SETREG       ;GET THE REG ADD
10695  057374  004076                           .WORD   RK5REG
10696  057376  004737  057710                   JSR     PC,SIZRK5       ;GET THE # OF DRIVES
10697  057402  000403                           BR      10$
10698
10699  057404  022626                   9$:     CMP     (SP)+,(SP)+     ;THERE WAS NO RK05
10700  057406  005037  177766                   CLR     @#CPUERR
10701
10702  057412  012737  057440  000004   10$:    MOV     #11$,@#4        ;IN CASE NON-EXISTENT REG
10703  057420  005777  124474                   TST     @UBEDB          ;TEST FOR UNIBUS EXERCISER
10704  057424  004737  057500                   JSR     PC,SETREG       ;GET THE REG ADD
10705  057430  004116                           .WORD   UBEREG
10706  057432  004737  057774                   JSR     PC,SIZUBE       ;GET THE #
10707  057436  000403                           BR      12$
10708
```

```
057440  022626                          11$:     CMP     (SP)+,(SP)+        ;THERE WAS NO UBE
057442  005037  177766                           CLR     @#CPUERR

057446  013737  057466  000004          12$:     MOV     SIZTM1,@#4         ;RESTORE 4
057454  104413                                   RESREG
057456  000207                                   RTS     PC

057460     000                          RS4DFL:  .BYTE   0
057461     000                          RP4DFL:  .BYTE   0
057462     000                          RH4DFL:  .BYTE   0
057463     000                          RK5DFL:  .BYTE   0
057464     000                          UBEDFL:  .BYTE   0
        057466                                   .EVEN

057466  000000                          SIZTM1:  .WORD   0
057470  000000                          SIZTM2:  .WORD   0
057472  000000                          SIZTM3:  .WORD   0
057474  000000                          SIZTM4:  .WORD   0
057476  000000                          SIZTM5:  .WORD   0

                                        ;THIS ROUTINE IS CALED BY A:
                                        ;                  JSR      PC,SETREG
                                        ;                  .WORD    DEVREG
                                        ;WHERE DEVREG IS THE STARTING ADDRESS OF
                                        ;A TABLE, WHICH IS TO CONTAIN THE ADDRESS OF
                                        ;A DEVICE'S CONTROL AND STATUS REGISTERS.
                                        ;THE TABLES ARE GENERATE HERE

057500  011637  057536                  SETREG:  MOV     (SP),SETMP
057504  062716  000002                           ADD     #2,(SP)
057510  104412                                   SAVREG
057512  017700  000020                           MOV     @SETMP,R0
057516  012001                                   MOV     (R0)+,R1
057520  011002                                   MOV     (R0),R2
057522  010220                          1$:      MOV     R2,(R0)+
057524  062702  000002                           ADD     #2,R2
057530  077104                                   SOB     R1,1$
057532  104413                                   RESREG
057534  000207                                   RTS     PC

057536  000000                          SETMP:   .WORD   0


                                        ;THIS ROUTINE IS CALLED, AFTER IT HAS BEEN
                                        ;DETERMINED IF THERE IS A RS04 CONTROLLER, TO SEE
                                        ;WHT DRIVES ARE AVAILABLE.

057540  012700  000010                  SIZRS4:  MOV     #10,R0

057544  012701  000001                           MOV     #1,R1
057550  005002                                   CLR     R2
057552  105037  057561                           CLRB    3$

057556  104426                          1$:      CALRS4
057560     001                          2$:      .BYTE   1
057561     000                          3$:      .BYTE   0                  ;DO A NOP FUNCTION
```

```
057562  000000                          .WORD   0               ;FOR EACH OF POSSIBLY
057564  000000                          .WORD   0               ;8 DRIVES
057566  000000                          .WORD   0
057570  000000                          .WORD   0
057572  000000                          .WORD   0
057574  000000                          .WORD   0

057576  005737  061004                  TST     RS4ER1
057602  001001                          BNE     4$
057604  050102                          BIS     R1,R2
057606  006301                  4$:     ASL     R1
057610  105237  057561                  INCB    3$
057614  077020                          SOB     R0,1$

057616  110237  057460                  MOVB    R2,RS4DFL
057622  000207                          RTS     PC

                                ;THIS ROUTINE IS CALLED TO DETERMINE WHAT RP04
                                ;DRIVES ARE ON THE CONTROLLER

057624  012700  000010          SIZRP4: MOV     #10,R0
057630  012701  000001                  MOV     #1,R1
057634  005002                          CLR     R2
057636  105037  057645                  CLRB    3$

057642  104427                  1$:     CALRP4                  ;DO A READ IN PRESET
057644     021                  2$:     .BYTE   21              ;FOR EACH OF UP TO
057645     000                  3$:     .BYTE   0               ;8 DRIVES.
057646  000000                          .WORD   0
057650  000000                          .WORD   0
057652  000000                          .WORD   0
057654  000000                          .WORD   0
057656  000000                          .WORD   0
057660  000000                          .WORD   0

057662  005737  060034                  TST     RP4ER1
057666  001001                          BNE     4$
057670  050102                          BIS     R1,R2
057672  006301                  4$:     ASL     R1
057674  105237  057645                  INCB    3$
057700  077020                          SOB     R0,1$

057702  110237  057461                  MOVB    R2,RP4DFL
057706  000207                          RTS     PC

                                ;DETERMINE WHAT RK05 DRIVES ARE AVAILABLE.

057710  012700  000010          SIZRK5: MOV     #10,R0
057714  012701  000001                  MOV     #1,R1
057720  005002                          CLR     R2

057722  105037  057731                  CLRB    3$


057726  104431                  1$:     CALRK5                  ;DO A DRIVE RESET
057730     015                          .BYTE   15              ;FOR EACH OF 8
```

```
057731     000                          3$:     .BYTE   0               ;POSSIBLE DRIVES.
057732  000000                                  .WORD   0
057734  000000                                  .WORD   0
057736  000000                                  .WORD   0
057740  000000                                  .WORD   0
057742  000000                                  .WORD   0
057744  000000                                  .WORD   0

057746  005737  061740                          TST     RK5ER1
057752  001001                                  BNE     4$
057754  050102                                  BIS     R1,R2
057756  006301                          4$:     ASL     R1
057760  105237  057731                          INCB    3$
057764  077020                                  SOB     R0,1$

057766  110237  057463                          MOVB    R2,RK5DFL
057772  000207                                  RTS     PC

                                        ;SET UP UBEDFL

057774  042777  000200  124124          SIZUBE: BIC     #BIT7,@UBECR1
060002  032777  000200  124116                  BIT     #BIT7,@UBECR1
060010  001403                                  BEQ     1$
060012  112737  000001  057464                  MOVB    #1,UBEDFL
060020  000207                          1$:     RTS     PC

                                        ;DETERMINE WHAT MASS BUS TESTER UNITS THERE ARE

060022  012737  000200  057462          SIZRH4: MOV     #BIT7,RH4DFL
060030  000207                                  RTS     PC

                                                .SBTTL  DEVICE HANDLERS
                                        ;;*****************************************************************
                                        ;*
                                        ;*      THE FOLLOWING SIX ROUTINES:
                                        ;*              RH4HAN
                                        ;*              RP4HAN
                                        ;*              RS4HAN
                                        ;*              UBEHAN
                                        ;*              RK5HAN
                                        ;*      ARE O/I AND BUS TESTER DEVICE HANDLERS.
                                        ;*      THEY ARE CALLED USING:
                                        ;*              TRAP TABLE CALL
                                        ;*      FUNCTION:.BYTE
                                        ;*              UNITNUM:.BYTE
                                        ;*      DISKADR1:.WORD
                                        ;*      DISKADR2:.WORD
                                        ;*      MEMADR1:.WORD
                                        ;*      MEMADR2:.WORD
                                        ;*      WORDCNT:.WORD
                                        ;*      VECTOR: .WORD
                                        ;*      RETURN:
                                        ;*A     WHERE TRAP TABLE CALL IS ONE OF:
                                        ;*              CALRH4
                                        ;*              CALRP4
                                        ;*              CALRS4
```

```
                            ;*              CALUBE
                            ;*              CALRK5
                            ;*B     FUNCTION IS THE PATTERN TO BE LOADED INTO THE
                            ;*      CONTROL REGISTER FUNCTION BITS, WITH EITHER
                            ;*      INTERRUPT ENABLED OR NOT.
                            ;*C     UNITNUM IS THE DRIVE NUMBER
                            ;*D     DISKADR1 AND DISKADR2 ARE THE DISK ADDRESS
                            ;*      SECTOR NUMBER
                            ;*E     MEMADR1 AND MEMADR2 ARE THE 22-BIT MEMORY
                            ;*      ADDRESS FOR THE TRANSFER.
                            ;*F     WORDCNT IS THE WORD COUNT A POSITIVE
                            ;*      NUMBER BETWEEN 0 AND 32K.
                            ;*G     VECTOR IS THE INTERRUPT HANDLER ROUTINE SPECIFIED
                            ;*      BY THE USER FOR AN INTERRUPT ENABLED FUNCTION.
                            ;*
                            ;*      WHEN THE HANDLER PROCESSES A CALL IT RETURNS
                            ;*      WITH THF 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
060032  000000              RP4FLG:.WORD    0
060034  000000              RP4ER1:.WORD    0               ;ERROR FLAGS.
060036  000000              RP4ER2:.WORD    0
060040  000000              RP4ER3:.WORD    0
060042  000000              RP4ER4:.WORD    0
060044  000000              RP4USE:.WORD    0
060046  000000              RP4TMP:.WORD    0
060050  000000              RP4FUN:.WORD    0
060052  000000              RP4UNI:.WORD    0
060054  000000              RP4DA1:.WORD    0
060056  000000              RP4DA2:.WORD    0
060060  000000              RP4MA1:.WORD    0
060062  000000              RP4MA2:.WORD    0
060064  000000              RP4WCT:.WORD    0
060066  000000              RP4VEC:.WORD    0
060070  000000              RP4TRK:.WORD    0
```

```
060072  000000                          RP4SEC:.WORD    0
060074  000000                          RP4CYL:.WORD    0

060076  005737  060032                  RP4HAN: TST     RP4FLG                  ;SEE IF THERE IS
060102  001402                                  BEQ     RP4H1                   ;ALREADY AN RP04 FUNCTION
060104  104000                                  ERROR                           ;IN PROGRESS. IF THERE
060106  000000                                  HALT                            ;IS ERROR> (SHOULD NEVER
060110  012737  000340  177776          RP4H1:  MOV     #340,@#PSW              ;HAPPEN.)
060116  011637  060046                          MOV     (SP),RP4TMP             ;RAISE THE PRIORITY
060122  062716  000016                          ADD     #16,(SP)
060126  104412                                  SAVREG                          ;GET AN ARGUMENT POINTER
060130  013700  060046                          MOV     RP4TMP,R0               ;RESET THE RETURN ADDRESS
060134  112037  060050                          MOVB    (R0)+,RP4FUN            ;FUNCTION
060140  112037  060052                          MOVB    (R0)+,RP4UNI            ;UNIT, DEVICE, NUMBER
060144  012037  060054                          MOV     (R0)+,RP4DA1            ;DISK ADDRESS
060150  012037  060056                          MOV     (R0)+,RP4DA2
060154  012037  060060                          MOV     (R0)+,RP4MA1            ;MEMORY ADDRESS
060160  012037  060062                          MOV     (R0)+,RP4MA2
060164  012037  060064                          MOV     (R0)+,RP4WCT            ;WORD COUNT
060170  012037  060066                          MOV     (R0)+,RP4VEC            ;INTERRUPT HANDLER ROUTINE
060174  005037  060034                          CLR     RP4ER1                  ;CLEAR THE ERROR
060200  005037  060036                          CLR     RP4ER2                  ;FLAGS
060204  005037  060040                          CLR     RP4ER3

060210  004737  060472                          JSR     PC,RP4S1                ;GO SET UP THE UNIT NUMBER
060214  004737  060542                          JSR     PC,RP4RDY               ;GET THE DEVICE READY.
060220  004737  060502                          JSR     PC,RP4S2                ;COMPUTE THE CYLINDER,
                                                                                ;TRACK AND SECTOR
060224  004737  060526                          JSR     PC,RP4S3                ;SET UP THE WORD COUNT

060230  013777  060052  123534          RP4H2:  MOV     RP4UNI,@RP4CS2          ;SET THE RP04 REGISTERS
060236  013777  060064  123520                  MOV     RP4WCT,@RP4WC           ;UP FOR THIS FUNCTION
060244  013777  060060  123514                  MOV     RP4MA1,@RP4BA
060252  013777  060062  123552                  MOV     RP4MA2,@RP4BAE
060260  013777  060056  123502                  MOV     RP4DA2,@RP4DA
060266  013777  060054  123522                  MOV     RP4DA1,@RP4DC
060274  013700  004136                          MOV     RP4V,R0                 ;SET UP THE INTERRUPT
060300  012720  060352                          MOV     #RP4H4,(R0)+            ;VECTOR
060304  012710  000340                          MOV     #340,(R0)
060310  013700  060050                          MOV     RP4FUN,R0               ;LOAD THE FUNCTION
060314  010037  060032                          MOV     R0,RP4FLG               ;AND GO
060320  110077  123436                          MOVB    R0,@RP4CS1
060324  032700  000100                          BIT     #BIT6,R0                ;SEE IF THE FUNCTION
060330  001402                                  BEQ     RP4H3                   ;WILL INTERRUPT WHEN
060332  104413                                  RESREG                          ;DONE. IF YES RETURN
060334  000002                                  RTI                             ;IF NOT INTERRUPTING
060336  004737  060366                  RP4H3:  JSR     PC,RP4H5                ;THEN WAIT FOR THE
060342  005037  060032                          CLR     RP4FLG                  ;FUNCTION TO FINISH.
060346  104413                                  RESREG                          ;THEN RETURN.
060350  000002                                  RTI

060352  005037  060032                  RP4H4:  CLR     RP4FLG                  ;WHEN THE INTERRUPT
060356  004737  060366                          JSR     PC,RP4H5                ;OCCURS CHECK FOR ERRORS
060362  000177  177500                          JMP     @RP4VEC                 ;AND GO TO THE SERVICE
                                                                                ;ROUTINE.

```

```
060366 010046                           RP4H5:  MOV     R0,-(SP)
060370 053777 060052 123374     RP4H51: BIS     RP4UNI,@RP4CS2
060376 017700 123360                            MOV     @RP4CS1,R0
060402 005700                                   TST     R0              ;SEE IF THE FUNCTION
060404 100023                                   BPL     RP4H6           ;WAS COMPLETED WITHOUT
060406 032700 060000                            BIT     #60000,R0       ;ERRORS.
060412 001420                                   BEQ     RP4H6
060414 017737 123352 060036                     MOV     @RP4CS2,RP4ER2  ;IF ERRORS OCCURRED SET
060422 017737 123346 060040                     MOV     @RP4DS,RF'ER3   ;THE INDICATORS
060430 017737 123342 060042                     MOV     @RP4RR1,RP4ER4
060436 012737 177777 060034                     MOV     #-1,RP4ER1
060444 004737 060764                            JSR     PC,RP4CLR       ;CLEAR THE CONTROL
060450 012600                                   MOV     (SP)+,R0
060452 000207                                   RTS     PC
060454 105700                           RP4H6:  TSTB    R0              ;WAIT FOR READY OR
060456 100344                                   BPL     RP4H51          ;ERROR
060460 105777 123310                            TSTB    @RP4DS
060464 100341                                   BPL     RP4H51
060466 012600                                   MOV     (SP)+,R0
060470 000207                                   RTS     PC

060472 042737 177770 060052     RP4S1:  BIC     #177770,RP4UNI  ;SET UP THE DRIVE NUMBER.
060500 000207                                   RTS     PC

060502 013701 060054                    RP4S2:  MOV     RP4DA1,R1       ;COMPUTE THE DISK
060506 005000                                   CLR     R0
060510 071027 000630                            DIV     #408.,R0
060514 010137 060054                            MOV     R1,RP4DA1
060520 005037 060056                            CLR     RP4DA2
060524 000207                                   RTS     PC

060526 005437 060064                    RP4S3:  NEG     RP4WCT          ;COMPUTE VALID WORD COUNT
060532 042737 177700 060062                     BIC     #177700,RP4MA2  ;AND MEMORY ADDRESS
060540 000207                                   RTS     PC

060542 012737 000040 060044     RP4RDY: MOV     #BIT5,RP4USE    ;CLEAR CONTROLLER AND
060550 053737 060052 060044                     BIS     RP4UNI,RP4USE
060556 013777 060044 123206                     MOV     RP4USE,@RP4CS2
060564 013777 060052 123200                     MOV     RP4UNI,@RP4CS2
060572 105777 123164                    1$:     TSTB    @RP4CS1         ;DRIVES
060576 100375                                   BPL     1$
060600 013777 060052 123164                     MOV     RP4UNI,@RP4CS2
060606 012777 000021 123146                     MOV     #21,@RP4CS1     ;INITIALIZE THE DRIVE
060614 017701 123142                    2$:     MOV     @RP4CS1,R1      ;BY DOING A NOP
060620 005701                                   TST     R1              ;WAIT FOR ERROR OR
060622 100434                                   BMI     4$              ;READY
060624 105701                                   TSTB    R1
060626 100372                                   BPL     2$

060630 017700 123140                    3$:     MOV     @RP4DS,R0       ;LOOK AT THE DRIVE
                                                                        ;STATUS
060634 032700 000400                            BIT     #BIT8,R0        ;DRIVE PRESENT?
060640 001430                                   BEQ     5$
060642 032700 000100                            BIT     #BIT6,R0        ;VOLUME VALID?
060646 001425                                   BEQ     5$
060650 032700 010000                            BIT     #BIT12,R0       ;ON LINE?
```

```
060654 001422                             BEQ     5$
060656 032700 040000                      BIT     #BIT14,R0         ;ANY ERRORS?
060662 001017                             BNE     5$
060664 032700 004000                      BIT     #BIT11,R0         ;WRITE LOCKED
060670 001014                             BNE     5$
060672 032700 001000                      BIT     #BIT9,  R0        ;PROGRAMMABLE DRIVE
060676 001011                             BNE     5$
060700 105700                             TSTB    R0                ;WAIT FOR DRIVE READY
060702 100344                             BPL     2$

060704 012777 010000 123102               MOV     #BIT12,@RP4OF     ;SET 16-BIT MODE
060712 000207                             RTS     PC                ;RETURN READY.
060714 032701 040000              4$:     BIT     #BIT14,R1         ;ATTENTION OR ERROR?
060720 001743                             BEQ     3$
060722 005726                     5$:     TST     (SP)+
060724 017737 123042 060036               MOV     @RP4CS2,RP4ER2    ;FLAG AND RECORD
060732 017737 123036 060040               MOV     @RP4DS,RP4ER3     ;ERROR
060740 017737 123032 060042               MOV     @RP4RR1,RP4ER4
060746 012737 177777 060034               MOV     #-1,RP4ER1
060754 004737 060764                      JSR     PC,RP4CLR         ;CLR THE CONTROLLER
060760 104413                             RESREG                    ;AND DRIVES.
060762 000002                             RTI                       ;RETURN

060764 013777 060044 123000       RP4CLR: MOV     RP4USE,@RP4CS2    ;CLR THE CONTROLLER
060772 105777 122764              1$:     TSTB    @RP4CS1           ;AND DRIVES.
060776 100375                             BPL     1$
061000 000207                             RTS     PC

                                          .SBTTL          RS04 DISK HANDLE
                                  ;RS04 DISK HANDLER

                                  ;REGISTERS USED IN RS4HAN
061002 000000                     RS4FLG:.WORD    0
061004 000000                     RS4ER1:.WORD    0                 ;ERROR FLAGS.
061006 000000                     RS4ER2:.WORD    0
061010 000000                     RS4ER3:.WORD    0
061012 000000                     RS4ER4:.WORD    0
061014 000000                     RS4USE:.WORD    0
061016 000000                     RS4TMP:.WORD    0
061020 000000                     RS4FUN:.WORD    0
061022 000000                     RS4UNI:.WORD    0
061024 000000                     RS4DA1:.WORD    0
061026 000000                     RS4DA2:.WORD    0
061030 000000                     RS4MA1:.WORD    0
061032 000000                     RS4MA2:.WORD    0
061034 000000                     RS4WCT:.WORD    0
061036 000000                     RS4VEC:.WORD    0
061040 000000                     RS4TRK:.WORD    0
061042 000000                     RS4SEC:.WORD    0
061044 000000                     RS4CYL:.WORD    0

061046 005737 061002              RS4HAN: TST     RS4FLG            ;SEE IF THERE ALREADY
061052 001402                             BEQ     RS4H1             ;IS AN RS04 FUNCTION
061054 104000                             ERROR                     ;IN PROGRESS. IF SO
061056 000000                             HALT                      ;ERROR. (SHOULD NEVER
061060 012737 000340 177776       RS4H1:  MOV     #340,@#PSW        ;HAPPEN.
```

```
061066 011637 061016                 MOV    (SP),RS4TMP
061072 062716 000016                 ADD    #16,(SP)
061076 104412                        SAVREG                 ;RAISE THE PRIORITY
061100 013700 061016                 MOV    RS4TMP,R0       ;GET A POINTER TO
061104 112037 061020                 MOVB   (R0)+,RS4FUN    ;FUNCTION
061110 112037 061022                 MOVB   (R0)+,RS4UNI    ;GET THE DRIVE NUMBER
061114 012037 061024                 MOV    (R0)+,RS4DA1    ;DISK ADDRESS
061120 012037 061026                 MOV    (R0)+,RS4DA2
061124 012037 061030                 MOV    (R0)+,RS4MA1    ;MEMORY ADDRESS
061130 012037 061032                 MOV    (R0)+,RS4MA2
061134 012037 061034                 MOV    (R0)+,RS4WCT    ;WORD COUNT
061140 012037 061036                 MOV    (R0)+,RS4VEC    ;INTERRUPT HANDLER ADDRESS
061144 005037 061004                 CLR    RS4ER1          ;CLEAR THE ERROR FLAGS
061150 005037 061006                 CLR    RS4ER2
061154 005037 061010                 CLR    RS4ER3
                                                                      ;SET UP UNIT (DRIVE) NUMBER
061160 004737 061434                 JSR    PC,RS4S1
061164 004737 061526                 JSR    PC,RS4RDY       ;INITIALIZE DRIVE AND
                                                            ;CONTROLLER
061170 004737 061444                 JSR    PC,RS4S2        ;COMPUTE TRACK AND SECTOR
061174 004737 061512                 JSR    PC,RS4S3        ;COMPUTE WORD COUNT.

061200 013777 061022 122526 PS4H2:   MOV    RS4UNI,@RS4CS2  ;SET UP THE CONTROL
061206 013777 061034 122512          MOV    RS4WCT,@RS4WC   ;AND DRIVE REGISTERS
061214 013777 061030 122506          MOV    RS4MA1,@RS4BA
061222 013777 061032 122524          MOV    RS4MA2,@RS4BAE
061230 013777 061024 122474          MOV    RS4DA1,@RS4DA
061236 013700 004134                 MOV    RS4V,R0
061242 012720 061314                 MOV    #RS4H4,(R0)+    ;SET THE INTERRUPT
061246 012710 000340                 MOV    #340,(R0)
061252 013700 061020                 MOV    RS4FUN,R0
061256 010037 061002                 MOV    R0,RS4FLG
061262 110077 122436                 MOVB   R0,@RS4CS1      ;LOAD THE FUNCTION AND GO.
061266 032700 000100                 BIT    #BIT6,R0        ;SEE IF AN INTERRUPT
061272 001402                        BEQ    RS4H3           ;IS TO BE EXPECTED.
061274 104413                        RESREG                 ;IF YES THEN RETURN
061276 000002                        RTI
                                                            ;IF NOT INTERRUPTING
061300 004737 061330        RS4H3:   JSR    PC,RS4H5        ;THEN WAIT FOR THE
061304 005037 061002                 CLR    RS4FLG          ;FUNCTION TO FINISH
061310 104413                        RESREG
061312 000002                        RTI

061314 005037 061002        RS4H4:   CLR    RS4FLG          ;WHEN THE INTERRUPT OCCURS.
061320 004737 061330                 JSR    PC,RS4H5        ;MAKE SURE THERE WERE
061324 000177 177506                 JMP    @RS4VEC         ;NO ERRORS BEFORE GOING
                                                            ;TO THE INTERRUPT
                                                            ;SERVICE ROUTINE.
061330 010046               RS4H5:   MOV    R0,-(SP)
061332 053777 061022 122374 RS4H51:  BIS    RS4UNI,@RS4CS2
061340 017700 122360                 MOV    @RS4CS1,R0
061344 005700                        TST    R0              ;SEE IF THE FUNCTION
061346 100023                        BPL    RS4H6           ;WAS COMPLETED WITHOUT
061350 032700 060000                 BIT    #60000,R0       ;ERRORS
061354 001420                        BEQ    RS4H6
061356 017737 122352 061006          MOV    @RS4CS2,RS4ER2  ;IF ERRORS OCCURRED
```

```
061364  017737  122346  061010              MOV     @RS4DS,RS4ER3     ;SET THE INDICATORS
061372  017737  122342  061012              MOV     @RS4ER,RS4ER4
061400  012737  177777  061004              MOV     #-1,RS4ER1
061406  004737  061720                      JSR     PC,RS4CLR         ;THEN CLEAR THE CONTROL
061412  012600                              MOV     (SP)+,R0
061414  000207                              RTS     PC                ;AND DRIVES
061416  105700                     RS4H6:   TSTB    R0
061420  100344                              BPL     RS4H51            ;WAIT FOR READY OR
061422  105777  122310                      TSTB    @RS4DS            ;ERROR
061426  100341                              BPL     RS4H51
061430  012600                              MOV     (SP)+,R0
061432  000207                              RTS     PC

061434  042737  177770  061022     RS4S1:   BIC     #177770,RS4UNI    ;SET UP DRIVE NUMBER
061442  000207                              RTS     PC

061444  013701  061024             RS4S2:   MOV     RS4DA1,R1         ;COMPUTE A DISK
061450  005000                              CLR     R0                ;ADDRESS
061452  071027  007000                      DIV     #3584.,R0
061456  005000                              CLR     R0
061460  071027  000100                      DIV     #100,R0
061464  010037  061040                      MOV     R0,RS4TRK
061470  010137  061044                      MOV     R1,RS4CYL
061474  000300                              SWAB    R0
061476  006200                              ASR     R0
061500  006200                              ASR     R0
061502  050001                              BIS     R0,R1
061504  010137  061024                      MOV     R1,RS4DA1
061510  000207                              RTS     PC

061512  005437  061034             RS4S3:   NEG     RS4WCT            ;COMPUTE A VALID WORD
061516  042737  177700  061032              BIC     #177700,RS4MA2    ;COUNT AND MEMORY
061524  000207                              RTS     PC                ;ADDRESS
061526  012737  000040  061014     RS4RDY:  MOV     #BIT5,RS4USE      ;CLEAR CONTROLLER AND DRIVES
061534  053737  061022  061014              BIS     RS4UNI,RS4USE
061542  013777  061014  122164              MOV     RS4USE,@RS4CS2
061550  013777  061022  122156              MOV     RS4UNI,@RS4CS2
061556  105777  122142             1$:      TSTB    @RS4CS1
061562  100375                              BPL     1$
061564  013777  061022  122142              MOV     RS4UNI,@RS4CS2
061572  012777  000001  122124              MOV     #1,@RS4CS1        ;INITIALIZE THE DRIVE
061600  017701  122120             2$:      MOV     @RS4CS1,R1        ;BY DOING A NOP.
061604  005701                              TST     R1
061606  100420                              BMI     4$
061610  105701                              TSTB    R1
061612  100372                              BPL     2$

061614  017700  122116             3$:      MOV     @RS4DS,R0         ;LOOK AT THE DRIVE STATUS
061620  032700  000400                      BIT     #BIT8,R0          ;DRIVE PRESENT?
061624  001414                              BEQ     5$
061626  032700  010000                      BIT     #BIT12,R0         ;ON LINE?
061632  001411                              BEQ     5$
061634  032700  004000                      BIT     #BIT11,R0         ;WRITE LOCKED?
061640  001006                              BNE     5$
061642  105700                              TSTB    R0                ;DRIVE READY?
061644  100355                              BPL     2$
```

```
061646  000207                                  RTS     PC
061650  032701  040000                  4$:     BIT     #BIT14,R1               ;ATTENTION OR ERROR?
061654  001757                                  BEQ     3$
061656  005726                          5$:     TST     (SP)+
061660  017737  122050  061006                  MOV     @RS4CS2,RS4ER2          ;FLAG AND RECORD THE
061666  017737  122044  061010                  MOV     @RS4DS,RS4ER3           ;ERROR
061674  017737  122040  061012                  MOV     @RS4ER,RS4ER4
061702  012737  177777  061004                  MOV     #-1,RS4ER1
061710  004737  061720                          JSR     PC,RS4CLR               ;CLR THE CONTROLLER
061714  104413                                  RESREG                          ;AND DRIVES AND RETURN.
061716  000002                                  RTI

061720  013777  061014  121776  RS4CLR: MOV     RS4USE,@RS4CS1          ;CLR THE CONTROLLER
061726  105777  121772          1$:     TSTB    @RS4CS1
061732  100375                                  BPL     1$
061734  000207                                  RTS     PC



                                                .SBTTL          RK05 DISK HANDLER
                                        ;RK05 DISK HANDLER

                                        ;REGISTERS USED IN RK5HAN
061736  000000                          RK5FLG:.WORD    0
061740  000000                          RK5ER1:.WORD    0                       ;ERROR FLAGS.
061742  000000                          RK5ER2:.WORD    0
061744  000000                          RK5ER3:.WORD    0
061746  000000                          RK5ER4:.WORD    0
061750  000000                          RK5USE:.WORD    0
061752  000000                          RK5TMP:.WORD    0
061754  000000                          RK5FUN:.WORD    0
061756  000000                          RK5UNI:.WORD    0
061760  000000                          RK5DA1:.WORD    0
061762  000000                          RK5DA2:.WORD    0
061764  000000                          RK5MA1:.WORD    0
061766  000000                          RK5MA2:.WORD    0
061770  000000                          RK5WCT:.WORD    0
061772  000000                          RK5VEC:.WORD    0
061774  000000                          RK5TRK:.WORD    0
061776  000000                          RK5SEC:.WORD    0
062000  000000                          RK5CYL:.WORD    0

062002  005737  061736                  RK5HAN: TST     RK5FLG                  ;SEE IF THERE IS ALREADY AN
062006  001402                                  BEQ     RK5H1                   ;RK05 FUNCTION IN PROGRESS
062010  104000                                  ERROR
062012  000000                                  HALT

062014  012737  000340  177776  RK5H1:  MOV     #340,@#PSW              ;RAISE THE PRIORITY
062022  011637  061752                          MOV     (SP),RK5TMP
062026  062716  000016                          ADD     #16,(SP)
062032  104412                                  SAVREG
062034  013700  061752                          MOV     RK5TMP,R0
062040  112037  061754                          MOVB    (R0)+,RK5FUN            ;GET THE ARGUMENTS.
062044  112037  061756                          MOVB    (R0)+,RK5UNI
062050  012037  061760                          MOV     (R0)+,RK5DA1
062054  012037  061762                          MOV     (R0)+,RK5DA2
062060  012037  061764                          MOV     (R0)+,RK5MA1
```

```
11269  062064  012037  061766                    MOV     (R0)+,RK5MA2
11270  062070  012037  061770                    MOV     (R0)+,RK5WCT
11271  062074  012037  061772                    MOV     (R0)+,RK5VEC
11272
11273  062100  005037  061740                    CLR     RK5ER1              ;CLR THE ERROR FLAGS
11274  062104  005037  061742                    CLR     RK5ER2
11275  062110  005037  061744                    CLR     RK5ER3
11276
11277  062114  004737  062364                    JSR     PC,RK5S1            ;SET UP THE DRIVE NUMBER
11278  062120  004737  062570                    JSR     PC,RK5RDY           ;GET THE DEVICE AND CONTROL
11279                                                                        ;READY
11280  062124  004737  062406                    JSR     PC,RK5S2            ;COMPUTE THE SURFACE
11281                                                                        ;CYLINDER AND SECTOR
11282                                                                        ;ADDRESS.
11283  062130  004737  062510                    JSR     PC,RK5S3            ;SET UP A WORD COUNT,
11284                                                                        ;THE UNIBUS MAP
11285                                                                        ;AND BUS ADDRESS.
11286
11287  062134  005077  121744          RK5H2:    CLR     @RK5CS1
11288  062140  013777  061756  121744            MOV     RK5UNI,@RK5DA       ;SET THE DEVICE REGISTERS
11289  062146  013777  061770  121732            MOV     RK5WCT,@RK5WC       ;TO DO THE FUNCTION
11290  062154  013777  061764  121726            MOV     RK5MA1,@RK5BA
11291  062162  053777  061766  121714            BIS     RK5MA2,@RK5CS1
11292  062170  053777  061760  121714            BIS     RK5DA1,@RK5DA
11293  062176  013700  004142                    MOV     RK5V,R0             ;LOAD THE INTERRUPT VECTOR
11294  062202  012720  062254                    MOV     #RK5H4,(R0)+
11295  062206  012710  000340                    MOV     #340,(R0)
11296  062212  013700  061754                    MOV     RK5FUN,R0
11297  062216  010037  061736                    MOV     R0,RK5FLG
11298  062222  050077  121656                    BIS     R0,@RK5CS1          ;LOAD THE FUNCTION AND
11299                                                                        ;GO
11300
11301  062226  032700  000100                    BIT     #BIT6,R0            ;SEE IF THE FUNCTION WILL
11302  062232  001402                            BEQ     RK5H3               ;INTERRUPT WHEN DONE.
11303  062234  104413                            RESREG                      ;IF YES RETURN
11304  062236  000002                            RTI
11305
11306  062240  004737  062302          RK5H3:    JSR     PC,RK5H5            ;IF THE FUNCTION WAS
11307  062244  005037  061736                    CLR     RK5FLG              ;NOT INTERRUPT ENABLED
11308  062250  104413                            RESREG                      ;WAIT FOR DONE OR ERROR.
11309  062252  000002                            RTI
11310
11311  062254  004737  062302          RK5H4:    JSR     PC,RK5H5            ;SEE IF THERE WERE ANY ERRORS.
11312  062260  005037  061736                    CLR     RK5FLG
11313  062264  012777  062300  121650            MOV     #1$,@RK5V
11314  062272  000230                            SPL     0
11315  062274  000177  177472                    JMP     @RK5VEC
11316  062300  000002                  1$:       RTI
11317
11318  062302  010046                  RK5H5:    MOV     R0,-(SP)
11319  062304  017700  121574          RK5H51:   MOV     @RK5CS1,R0          ;SEE IF ANY ERROR OCCURRED
11320  062310  005700                            TST     R0
11321  062312  100015                            BPL     RK5H6
11322  062314  017737  121562  061742            MOV     @RK5ER,RK5ER2       ;IF YES, FLAG THE ERROR
11323  062322  017737  121552  061744            MOV     @RK5DS,RK5ER3       ;AND SAVE THE STATUS
11324  062330  012737  177777  061740            MOV     #-1,RK5ER1
```

```
062336  004737  062730                          JSR     PC,RK5CLR
062342  012600                                  MOV     (SP)+,R0
062344  000207                                  RTS     PC

062346  105700                        RK5H5:    TSTB    R0                      ;WAIT FOR DONE OR
062350  100355                                  BPL     RK5H51                  ;ERROR
062352  105777  121522                          TSTB    @RK5DS
062356  100352                                  BPL     RK5H51
062360  012600                                  MOV     (SP)+,R0
062362  000207                                  RTS     PC

062364  013700  061756                RK5S1:    MOV     RK5UNI,R0
062370  072027  000015                          ASH     #13.,R0
062374  042700  017777                          BIC     #017777,R0
062400  010037  061756                          MOV     R0,RK5UNI
062404  000207                                  RTS     PC

062406  013701  061760                RK5S2:    MOV     RK5DA1,R1               ;COMPUTE THE CYLINDER
062412  005000                                  CLR     R0                      ;SURFACE AND SECTOR
062414  071027  011100                          DIV     #4672.,R0               ;DISK ADDRESS
062420  005000                                  CLR     R0
062422  071027  000030                          DIV     #24.,R0
062426  010002                                  MOV     R0,R2
062430  005000                                  CLR     R0
062432  071027  000014                          DIV     #12.,R0
062436  010237  062000                          MOV     R2,RK5CYL
062442  010137  061776                          MOV     R1,RK5SEC
062446  010037  061774                          MOV     R0,RK5TRK
062452  072227  000005                          ASH     #5,R2
062456  042702  160037                          BIC     #160037,R2
062462  072027  000004                          ASH     #4,R0
062466  042700  177757                          BIC     #177757,R0
062472  042701  177760                          BIC     #177760,R1
062476  050100                                  BIS     R1,R0
062500  050200                                  BIS     R2,R0
062502  010037  061760                          MOV     R0,RK5DA1
062506  000207                                  RTS     PC

062510  005437  061770                RK5S3:    NEG     RK5WCT                  ;COMPUTE A VALID
                                                                                ;WORD COUNT AND
062514  013700  061764                          MOV     RK5MA1,R0               ;SET THE UB MAP
062520  013701  061766                          MOV     RK5MA2,R1               ;REGISTERS
062524  042701  177700                          BIC     #177700,R1
062530  012702  170300                          MOV     #MAPL20,R2
062534  012703  000010                          MOV     #10,R3
062540  010022                        1$:       MOV     R0,(R2)+
062542  010122                                  MOV     R1,(R2)+
062544  062700  020000                          ADD     #20000,R0
062550  005501                                  ADC     R1
062552  077306                                  SOB     R3,1$
062554  012737  000040  061766                  MOV     #40,RK5MA2
062562  005037  061764                          CLR     RK5MA1
062566  000207                                  RTS     PC

062570  053777  061756  121314        RK5RDY:   BIS     RK5UNI,@RK5DA           ;DO A CONTROL CLEAR
062576  012777  000001  121300                  MOV     #1,@RK5CS1              ;FUNCTION
```

```
062604  105777  121274          1$:     TSTB    @RK5CS1
062610  100375                          BPL     1$

062612  053777  061756  121272          BIS     RK5UNI,@RK5DA   ;DO A DRIVE CLEAR
062620  012777  000015  121256          MOV     #15,@RK5CS1     ;FUNCTION

062626  017701  121252          2$:     MOV     @RK5CS1,R1      ;WAIT FOR DONE OR
062632  100420                          BMI     5$              ;ERROR.
062634  105701                          TSTB    R1
062636  100373                          BPL     2$

062640  017701  121234          3$:     MOV     @RK5DS,R1
062644  032701  000040                  BIT     #BIT5,R1        ;WRITE ENABLED?
062650  001011                          BNE     5$
062652  005777  121224                  TST     @RK5ER
062656  100406                          BMI     5$
062660  105701                          TSTB    R1
062662  100366                          BPL     3$
062664  032701  000100                  BIT     #BIT6,R1
062670  001763                          BEQ     3$
062672  000207                  4$:     RTS     PC

062674  005726                  5$:     TST     (SP)+
062676  017737  121200  061742          MOV     @RK5ER,RK5ER2
062704  017737  121170  061744          MOV     @RK5DS,RK5ER3
062712  012737  177777  061740          MOV     #-1,RK5ER1
062720  004737  062730                  JSR     PC,RK5CLR
062724  104413                          RESREG
062726  000002                          RTI

062730  005077  121156          RK5CLR: CLR     @RK5DA          ;RESET THE CONTROLLER
062734  012777  000001  121142          MOV     #1,@RK5CS1      ;BY DOING A CONTROL
062742  105777  121136          1$:     TSTB    @RK5CS1         ;CLEAR FUNCTION
062746  100375                          BPL     1$
062750  000207                          RTS     PC

                                        .SBTTL          UNIBUS EXERCISER HANDLER
                                ;UNIBUS EXERCISER HANDLER

                                ;REGISTERS USED IN UBEHAN
062752  000000                  UBEFLG:.WORD    0
062754  000000                  UBEER1:.WORD    0               ;ERROR FLAGS.
062756  000000                  UBEER2:.WORD    0
062760  000000                  UBEER3:.WORD    0
062762  000000                  UBEER4:.WORD    0
062764  000000                  UBEUSE:.WORD    0
062766  000000                  UBETMP:.WORD    0
062770  000000                  UBEFUN:.WORD    0
062772  000000                  UBEUNI:.WORD    0
062774  000000                  UBEDA1:.WORD    0
062776  000000                  UBEDA2:.WORD    0
063000  000000                  UBEMA1:.WORD    0
063002  000000                  UBEMA2:.WORD    0
063004  000000                  UBEWCT:.WORD    0
063006  000000                  UBEVEC:.WORD    0
063010  000000                  UBETRK:.WORD    0
```

```
063012  000000                          UBESEC:.WORD    0
063014  000000                          UBECYL:.WORD    0

063016  005737  062752                  UBEHAN: TST     UBEFLG                  ;SEE IF THERE IS ALREADY
063022  001402                                  BEQ     UBEH1                   ;A UNIBUS EXERCISER FUNCTION
063024  104000                                  ERROR                           ;IN PROGRESS. IF THERE
063026  000000                                  HALT                            ;IS ERROR. (SHOULD NEVER HAPPEN)

063030  012737  000340  177776          UBEH1:  MOV     #340,@#PSW              ;RAISE THE PRIORITY
063036  011637  062766                          MOV     (SP),UBETMP             ;GET AN ARGUMENT POINTER
063042  062716  000016                          ADD     #16,(SP)
063046  104412                                  SAVREG
063050  013700  062766                          MOV     UBETMP,R0               ;RESET THE RETURN ADDRESS

063054  012037  062770                          MOV     (R0)+,UBEFUN            ;GET THE ARGUMENTS.
063060  012037  062774                          MOV     (R0)+,UBEDA1
063064  012037  062776                          MOV     (R0)+,UBEDA2
063070  012037  063000                          MOV     (R0)+,UBEMA1
063074  012037  063002                          MOV     (R0)+,UBEMA2
063100  012037  063004                          MOV     (R0)+,UBEWCT
063104  012037  063006                          MOV     (R0)+,UBEVEC
063110  005037  062754                          CLR     UBEER1                  ;CLEAR THE ERROR FLAGS
063114  005037  062756                          CLR     UBEER2
063120  005037  062760                          CLR     UBEER3
063124  004737  063406                          JSR     PC,UBERDY
063130  004737  063332                          JSR     PC,UBES1                          ;GO SET UP THE BUS
                                                                                ;ADDRESS AND UB MAP


063134  013777  063004  120760          UBEH2:  MOV     UBEWCT,@UBECC           ;SET THE DEVICE
063142  012777  060000  120754                  MOV     #60000,@UBEBA           ;REGISTERS
063150  053777  063002  120754                  BIS     UBEMA2,@UBECR2
063156  013777  062776  120734                  MOV     UBEDA2,@UBEDB
063164  013700  004144                          MOV     UBEV,R0
063170  012720  063242                          MOV     #UBEH4,(R0)+
063174  012710  000340                          MOV     #340,(R0)
063200  013700  062770                          MOV     UBEFUN,R0
063204  010037  062752                          MOV     R0,UBEFLG
063210  010077  120712                          MOV     R0,@UBECR1              ;LOAD THE FUNCTION
063214  032700  000100                          BIT     #BIT6,R0                ;SEE IF THE FUNCTION
063220  001402                                  BEQ     UBEH3                   ;IS INTERRUPT ENABLED
063222  104413                                  RESREG                          ;IF YES RETURN
063224  000002                                  RTI

063226  004737  063256                  UBEH3:  JSR     PC,UBEH5                ;IF NOT INTERRUPT ENABLED
063232  005037  062752                          CLR     UBEFLG                  ;WAIT FOR DONE OR
063236  104413                                  RESREG                          ;ERROR
063240  000002                                  RTI

063242  005037  062752                  UBEH4:  CLR     UBEFLG                  ;WHEN THE INTERRUPT
063246  004737  063256                          JSR     PC,UBEH5                ;OCCURS SEE IF ANY ERRORS
063252  000177  177530                          JMP     @UBEVEC                 ;OCCURRED

063256  010046                          UBEH5:  MOV     R0,-(SP)
063260  017700  120642                  UBEH51: MOV     @UBECR1,R0              ;WAIT FOR DONE OR
063264  005700                                  TST     R0                      ;ERROR
063266  100015                                  BPL     UBEH6
```

```

063270 017737 120632 062756          MOV     @UBECR1,UBEER2
063276 017737 120630 062760          MOV     @UBECR2,UBEER3
063304 012737 177777 062754          MOV     #-1,UBEER1
063312 004737 063462                 JSR     PC,UBCLR
063316 012600                        MOV     (SP)+,R0
063320 000207                        RTS     PC

063322 105700               UBEH6:   TSTB    R0
063324 100355                        BPL     UBEH51
063326 012600                        MOV     (SP)+,R0
063330 000207                        RTS     PC

063332 013700 063000        UBES1:   MOV     UBEMA1,R0       ;SET UP THE BUS ADDRESS
063336 013701 063002                 MOV     UBEMA2,R1       ;AND UB MAPPING BOX
063342 042701 177700                 BIC     #177700,R1
063346 012702 170214                 MOV     #MAPL03,R2

063352 010022               1$:      MOV     R0,(R2)+
063354 010122                        MOV     R1,(R2)+
063356 062700 020000                 ADD     #20000,R0
063362 005501                        ADC     R1

063364 005037 063002                 CLR     UBEMA2
063370 005037 063000                 CLR     UBEMA1
063374 005137 063004                 COM     UBEWCT
063400 005237 063004                 INC     UBEWCT
063404 000207                        RTS     PC

063406 005077 120516        UBERDY:  CLR     @UBECLR         ;TRY TO GET DEVICE
                                                             ;READY

063412 017700 120510        1$:      MOV     @UBECR1,R0
063416 100403                        BMI     2$
063420 105700                        TSTB    R0
063422 100373                        BPL     1$
063424 000207                        RTS     PC

063426 005726               2$:      TST     (SP)+
063430 017737 120472 062756          MOV     @UBECR1,UBEER2
063436 017737 120470 062760          MOV     @UBECR2,UBEER3
063444 012737 177777 062760          MOV     #-1,UBEER3
063452 004737 063462                 JSR     PC,UBCLR
063456 104413                        RESREG
063460 000002                        RTI

063462 005077 120442        UBCLR:   CLR     @UBECLR         ;CLEAR THE DEVICE.
063466 105777 120434        1$:      TSTB    @UBECR1
063472 100375                        BPL     1$
063474 000207                        RTS     PC

                                      .SBTTL        MASS BUS TESTER HANDLER
                            ;THIS CODE IS FOR HANDLING THE MASS BUS
                            ;TESTED DEVICE.

                            ;REGISTERS USED IN RH4HAN
063476 000000               RH4FLG:.WORD      0
```

```
063500  000000                          RH4ER1:.WORD    0                       ;ERROR FLAGS.
063502  000000                          RH4ER2:.WORD    0
063504  000000                          RH4ER3:.WORD    0
063506  000000                          RH4ER4:.WORD    0
063510  000000                          RH4USE:.WORD    0
063512  000000                          RH4TMP:.WORD    0
063514  000000                          RH4FUN:.WORD    0
063516  000000                          RH4UNI:.WORD    0
063520  000000                          RH4DA1:.WORD    0
063522  000000                          RH4DA2:.WORD    0
063524  000000                          RH4MA1:.WORD    0
063526  000000                          RH4MA2:.WORD    0
063530  000000                          RH4WCT:.WORD    0
063532  000000                          RH4VEC:.WORD    0
063534  000000                          RH4TRK:.WORD    0
063536  000000                          RH4SEC:.WORD    0
063540  000000                          RH4CYL:.WORD    0

063542  005737  063476                  RH4HAN: TST     RH4FLG                  ;SEE IF A FUNCTION
063546  001402                                  BEQ     RH4H1                   ;IS ALREADY ACTIVE IF
063550  104000                                  ERROR                           ;SO ERROR.
063552  000000                                  HALT

063554  012777  000340  114214          RH4H1:  MOV     #340,@PSW               ;RAISE THE PRIORITY
063562  011637  063512                          MOV     (SP),RH4TMP
063566  062716  000016                          ADD     #16,(SP)
063572  104412                                  SAVREG
063574  013700  063512                          MOV     RH4TMP,R0               ;RESET THE RETURN
063600  112037  063514                          MOVB    (R0)+,RH4FUN
063604  112037  063516                          MOVB    (R0)+,RH4UNI
063610  012037  063520                          MOV     (R0)+,RH4DA1
063614  012037  063522                          MOV     (R0)+,RH4DA2
063620  012037  063524                          MOV     (R0)+,RH4MA1
063624  012037  063526                          MOV     (R0)+,RH4MA2
063630  012037  063530                          MOV     (R0)+,RH4WCT
063634  011037  063532                          MOV     (R0),RH4VEC
063640  005037  063500                          CLR     RH4ER1                  ;CLEAR THE ERROR FLAGS
063644  005037  063502                          CLR     RH4ER2
063650  005037  063504                          CLR     RH4ER3
063654  004737  064134                          JSR     PC,RH4S1                ;SET UP THE UNIT NUMBER
063660  004737  064160                          JSR     PC,RH4RDY               ;GET THE UNIT READY
063664  004737  064144                          JSR     PC,RH4S2

063670  013777  063516  120152          RH4H2:  MOV     RH4UNI,@RH4CS2          ;SFT THE CONTROL REGISTERS
063676  013777  063530  120136                  MOV     RH4WCT,@RH4WC           ;AND DEVICE REGISTERS
063704  013777  063524  120132                  MOV     RH4MA1,@RH4BA
063712  013777  063526  120152                  MOV     RH4MA2,@RH4AE
063720  013777  063520  120132                  MOV     RH4DA1,@RH4DR
063726  012777  004000  120130                  MOV     #4000,@RH4MR1
063734  000240                                  NOP
063736  013700  004140                          MOV     RH4V,R0                 ;VECTOR
063742  012720  064014                          MOV     #RH4H4,(R0)+
063746  012710  000340                          MOV     #340,(R0)
063752  013700  063514                          MOV     RH4FUN,R0
063756  010037  063476                          MOV     R0,RH4FLG               ;LOAD THE FUNCTION AND
063762  110077  120052                          MOVB    R0,@RH4CS1              ;GO
```

```
11605 063766 032700 000100                   BIT     #BIT6,R0            ;SEE IF THIS FUNCTION
11606 063772 001402                          BEQ     RH4H3               ;WILL INTERRUPT WHEN DONE
11607 063774 104413                          RESREG                      ;IF YES RETURN TO CALL
11608 063776 000002                          RTI
11609
11610 064000 004737 064030         RH4H3:    JSR     PC,RH4H5            ;IF NOT INTERRUPT
11611 064004 005037 063476                   CLR     RH4FLG              ;ENABLED WAIT FOR
11612 064010 104413                          RESREG                      ;THE FUNCTION TO
11613 064012 000002                          RTI                         ;FINISH THEN RETURN.
11614
11615 064014 005037 063476         RH4H4:    CLR     RH4FLG              ;WHEN THE INTERRUPT
11616 064020 004737 064030                   JSR     PC,RH4H5            ;OCCURS CHECKS FOR
11617 064024 000177 177502                   JMP     @RH4VEC             ;ERRORS. THEN GO TO THE
11618                                                                      ;SPECIFIED SERVICE
11619                                                                      ;ROUTINE
11620
11621 064030 010046                RH4H5:    MOV     R0,-(SP)
11622 064032 053777 063516 120010 RH4H51:   BIS     RH4UNI,@RH4CS2
11623 064040 017700 117774                   MOV     @RH4CS1,R0          ;SEE IF THE FUNCTION
11624 064044 005700                          TST     R0                  ;WAS COMPLETED WITHOUT
11625 064046 100023                          BPL     RH4H6               ;ERRORS.
11626 064050 032700 060000                   BIT     #60000,R0
11627 064054 001420                          BEQ     RH4H6
11628 064056 017737 117766 063502            MOV     @RH4CS2,RH4ER2      ;IF ERRORS OCCURRED
11629 064064 017737 117762 063504            MOV     @RH4ST,RH4ER3       ;SAVE STATUS AND SET
11630 064072 017737 117756 063506            MOV     @RH4ER,RH4ER4
11631 064100 012737 177777 063500            MOV     #-1,RH4ER1          ;ERROR FLAGS.
11632 064106 004737 064352                   JSR     PC,RH4CLR
11633 064112 012600                          MOV     (SP)+,R0
11634 064114 000207                          RTS     PC
11635
11636 064116 105700                RH4H6:    TSTB    R0                  ;WAIT FOR READY OR
11637 064120 100344                          BPL     RH4H51              ;ERROR
11638 064122 105777 117724                   TSTB    @RH4ST
11639 064126 100341                          BPL     RH4H51
11640 064130 012600                          MOV     (SP)+,R0
11641 064132 000207                          RTS     PC
11642
11643 064134 042737 177770 063516 RH4S1:    BIC     #177770,RH4UNI      ;SET UP THE DRIVE NUMBER
11644 064142 000207                          RTS     PC
11645
11646 064144 012737 000000 063522 RH4S2:    MOV     #0,RH4DA2           ;FOR DEBUG.
11647 064152 005437 063530                   NEG     RH4WCT              ;SET UP WORD COUNT
11648 064156 000207                          RTS     PC
11649
11650 064160 012737 000040 063510 RH4RDY:   MOV     #BIT5,RH4USE        ;CLR THE CONTROLLER
11651 064166 053737 063516 063510            BIS     RH4UNI,RH4USE
11652 064174 013777 063510 117646            MOV     RH4USE,@RH4CS2
11653 064202 013777 063516 117640            MOV     RH4UNI,@RH4CS2
11654 064210 105777 117624          1$:       TSTB    @RH4CS1 ;AND DRIVES
11655 064214 100375                          BPL     1$
11656 064216 013777 063516 117624            MOV     RH4UNI,@RH4CS2      ;DO A NOP FUNCTION
11657 064224 012777 000001 117606            MOV     #1,@RH4CS1          ;TO INITIALIZE THE
11658                                                                      ;DRIVE
11659 064232 017701 117602          2$:       MOV     @RH4CS1,R1          ;WAIT FOR READY OR ERROR.
11660 064236 005701                          TST     R1
```

```
064240  100420                                    BMI     4$
064242  105701                                    TSTB    R1
064244  100372                                    BPL     2$

064246  017700  117600            3$:             MOV     @RH4ST,R0            ;LOOK AT THE UNIT STATUS
064252  032700  000400                            BIT     #BIT8,R0             ;UNIT PRESENT?
064256  001414                                    BEQ     5$
064260  032700  010000                            BIT     #BIT12,R0            ;ON LINE?
064264  001411                                    BEQ     5$
064266  032700  040000                            BIT     #BIT14,R0            ;ANY ERRORS?
064272  001006                                    BNE     5$
064274  105700                                    TSTB    R0                   ;WAIT FOR UNIT READY
064276  100355                                    BPL     2$
064300  000207                                    RTS     PC

064302  032701  040000            4$:             BIT     #BIT14,R1            ;ATTENTION OR ERROR
064306  001757                                    BEQ     3$
064310  005726                    5$:             TST     (SP)+                ;FLAG AND RECORD ERROR
064312  017737  117532  063502                    MOV     @RH4CS2,RH4ER2
064320  017737  117526  063504                    MOV     @RH4ST,RH4ER3
064326  017737  117522  063506                    MOV     @RH4ER,RH4ER4
064334  012737  177777  063500                    MOV     #-1,RH4ER1
064342  004737  064352                            JSR     PC,RH4CLR
064346  104413                                    RESREG
064350  000002                                    RTI

064352  013777  063510  117470    RH4CLR: MOV     RH4USE,@RH4CS2       ;CLR THE CONTROLLER
064360  105777  117454            1$:     TSTB    @RH4CS1              ;AND DRIVES.
064364  100375                                    BPL     1$
064366  000207                    TSTDT1: RTS     PC
064370  001000                                    .BLKW   512.
                                  ;SPECIAL MESSAGES:

066370  041536  000200            CONCMS: .ASCIZ  '^C'<CRLF>

066374  047515  044516  047524    MMESRS: .ASCIZ  'MONITOR (OR LOADER) RESTORED!'<CRLF>
066402  020122  047450  020122
066410  047514  042101  051105
066416  020051  042522  052123
066424  051117  042105  100041
066432     000

066433     200  047520  042527    POWERM: .ASCIZ  <CRLF>'POWER FAILURE, PROGRAM RESTARTING'<CRLF><CRLF>
066440  020122  040506  046111
066446  051125  026105  050040
066454  047522  051107  046501
066462  051040  051505  040524
066470  052122  047111  100107
066476  000200

066500  000011                    $TAB:   .ASCIZ  <TAB>

066502  042600  050130  041505    MTA5:   .ASCII  <CRLF>'EXPECTED DATA:'<CRLF>
066510  042524  020104  040504
066516  040524  100072
066522  051107  052517  020120            .ASCIZ  'GROUP 0.GROUP 1.MEM EV.'<TAB>'MEM ODD.'<CRLF>
```

```
066530  027060  051107  052517
066536  020120  027061  042515
066544  020115  053105  004456
066552  042515  020115  042117
066560  027104  000200

066564  042200  052101  020101  MTA11:  .ASCII  <CRLF>'DATA WRITTEN.'<TAB>'TEST ADDR.'<TAB>'ERROR REG.'<CRLF>
066572  051127  052111  042524
066600  027116  052011  051505
066606  020124  042101  051104
066614  004456  051105  047522
066622  020122  042522  027107
066630     200

066631     040  047111  00004C  MTA17:  .ASCIZ  ' IN '

066636  054105  042520  052103  MTB17:  .ASCIZ  'EXPECTED DATA:'<CRLF>
066644  042105  042040  052101
066652  035101  000200

066656  054502  042524  004456  MTC17:  .ASCIZ  'BYTE.'<TAB>
066664     000

066665     127  051117  027104  MTA20:  .ASCIZ  'WORD.'<TAB>
066672  000011

066674  054105  042520  052103  MTA21:  .ASCII  'EXPECTED DATA:'<CRLF>
066702  042105  042040  052101
066710  035101     200
066713     110  052111  020123          .ASCIZ  'HITS IN GROUP 0.'<TAB>'/'<TAB>'HITS IN GROUP 1. '<CRLF>
066720  047111  043440  047522
066726  050125  030040  004456
066734  004457  044510  051524
066742  044440  020116  051107
066750  052517  020120  027061
066756  100040     000

        066631                  MTB21=MTA17

066761     200  042524  052123  MTA43:  .ASCII  <CRLF>'TEST ADDRESS.'<TAB>'ERROR ADRS REG.'<TAB>
066766  040440  042104  042522
066774  051523  004456  051105
067002  047522  020122  042101
067010  051522  051040  043505
067016  004456
067020  051105  047522  020122          .ASCIZ  'ERROR REG.'<CRLF>
067026  042522  027107  000200

067034  053600  047522  042524  MTA45:  .ASCIZ  <CRLF>'WROTE. 377'<TAB>'IN BYTE. '
067042  020056  033463  004467
067050  047111  041040  052131
067056  027105  000040

067062  051200  040505  020104  MTB45:  .ASCIZ  <CRLF>'READ DATA. '
067070  040504  040524  020056
067076     000
```

```
067077     011  047111  053440  MTC45:  .ASCIZ  <TAB>'IN WORD. '
067104  051117  027104  000040

067112  053600  047522  042524  MTA50:  .ASCIZ  <CRLF>'WROTE. 000'<TAB>'IN BYTE. '
067120  020056  030060  004460
067126  047111  041040  052131
067134  027105  000040

067140  042600  052116  051105  PDMSG1: .ASCII  <CRLF>'ENTERING CACHE ADDRESS MEMORY POWER UP '
067146  047111  020107  040503
067154  044103  020105  042101
067162  051104  051505  020123
067170  042515  047515  054522
067176  050040  053517  051105
067204  052440  020120
067210  047111  040526  044514          .ASCII  'INVALIDATOR TEST.'<CRLF>
067216  040504  047524  020122
067224  042524  052123  100056
067232  046120  040505  042523          .ASCII  'PLEASE GO THROUGH A POWER DOWN, POWER UP '
067240  043440  020117  044124
067246  047522  043525  020110
067254  020101  047520  042527
067262  020122  047504  047127
067270  020054  047520  042527
067276  020122  050125     040
067303     123  050505  042525          .ASCIZ  'SEQUENCE.'<CRLF>
067310  041516  027105  000200

067316  041600  041501  042510  PDMSG2: .ASCII  <CRLF>'CACHE ADDRESS MEMORY POWER UP INVALIDATOR'
067324  040440  042104  042522
067332  051523  046440  046505
067340  051117  020131  047520
067346  042527  020122  050125
067354  044440  053116  046101
067362  042111  052101  051117
067370  052040  051505  020124          .ASCIZ  ' TEST DID NOT FAIL.'<CRLF>
067376  044504  020104  047516
067404  020124  040506  046111
067412  100056     000

067415     105  051122  051117  ADRNG:  .ASCII  'ERROR ADDRESS REGISTER NEEDED FOR TEST.'<CRLF>'BUT IT HAS BEEN '
067422  040440  042104  042522
067430  051523  051040  043505
067436  051511  042524  020122
067444  042516  042105  042105
067452  043040  051117  052040
067460  051505  026124  041200
067466  052125  044440  020124
067474  040510  020123  042502
067502  047105     040
067505     106  040514  043507          .ASCIZ  'FLAGGED AS BAD''
067512  042105  040440  020123
067520  040502  020504     000

067525     105  051122  051117  ERRNG:  .ASCII  'ERROR REGISTER NEEDED FOR TEST.'<CRLF>'BUT IT HAS BEEN '
```

```
067532  051040  043505  051511
067540  042524  020122  042516
067546  042105  042105  043040
067554  051117  052040  051505
067562  026124  041200  052125
067570  044440  020124  040510
067576  020123  042502  047105
067604     040
067605     106  040514  043507                  .ASCIZ  'FLAGGED AS BAD.'
067612  042105  040440  020123
067620  040502  020504     000

067625     103  047117  051124   CNRNG:  .ASCII  'CONTROL REGISTER NEEDED FOR TEST.'<CRLF>'BUT IT HAS BEEN '
067632  046117  051040  043505
067640  051511  042524  020122
067646  042516  042105  042105
067654  043040  051117  052040
067662  051505  026124  041200
067670  052125  044440  020124
067676  040510  020123  042502
067704  047105     040
067707     106  040514  043507                  .ASCIZ  'FLAGGED AS BAD.'
067714  042105  040440  020123
067722  040502  020504     000
067727     115  044501  052116   MNRNG:  .ASCII  'MAINTENANCE REGISTER NEEDED FOR TEST.'<CRLF>'BUT IT HAS BEEN '
067734  047105  047101  042503
067742  051040  043505  051511
067750  042524  020122  042516
067756  042105  042105  043040
067764  051117  052040  051505
067772  026124  041200  052125
070000  044440  020124  040510
070006  020123  042502  047105
070014     040
070015     106  040514  043507                  .ASCIZ  'FLAGGED AS BAD!'
070022  042105  040440  020123
070030  040502  020504     000

070035     110  052111  046457   HMRNG:  .ASCII  'HIT/MISS REGISTER NEEDED FOR TEST.'<CRLF>'BUT IT HAS BEEN '
070042  051511  020123  042522
070050  044507  052123  051105
070056  047040  042505  042504
070064  020104  047506  020122
070072  042524  052123  100054
070100  052502  020124  052111
070106  044040  051501  041040
070114  042505  020116
070120  046106  043501  042507                  .ASCIZ  'FLAGGED AS BAD!'
070126  020104  051501  041040
070134  042101  000041

070140  040600  042104  042522   MTA77:  .ASCIZ  <CRLF>'ADDRESS:  '
070146  051523  020072  000040

070154  051440  047510  046125   MTB77:  .ASCIZ  ' SHOULD HAVE BEEN A HIT IN GROUP '
070162  020104  040510  042526
```

```
070170  041040  042505  020116
070176  020101  044510  020124
070204  047111  043440  047522
070212  050125  000040

070216  043101  042524  020122  MTC77:  .ASCIZ  'AFTER REFERENCING'<CRLF>'ADDRESS:  '
070224  042522  042506  042522
070232  041516  047111  100107
070240  042101  051104  051505
070246  035123  020040     000

070253     040  044127  046111  MTD77:  .ASCIZ  ' WHILE FORCING SELECTION OF GROUP '
070260  020105  047506  041522
070266  047111  020107  042523
070274  042514  052103  047511
070302  020116  043117  043440
070310  047522  050125  000040

070316  040600  051122  051117  MTA101: .ASCII  <CRLF>'ARROR ADRS REG.'<TAB>'ERROR REG.'<TAB>
070324  040440  051104  020123
070332  042522  027107  042411
070340  051122  051117  051040
070346  043505  004456
070352  054105  042520  052103          .ASCIZ  'EXPECTED ERR.'<TAB>'PATTERN PUT IN MAINT REG.'<CRLF>
070360  042105  042440  051122
070366  004456  040520  052124
070374  051105  020116  052520
070402  020124  047111  046440
070410  044501  052116  051040
070416  043505  100056     000

070423     200  043101  042524  MTA120: .ASCIZ  <CRLF>'AFTER 2ND CYCLE READ  '
070430  020122  047062  020104
070436  054503  046103  020105
070444  042522  042101  020040
070452     000

070453     200  043101  042524  MTB120: .ASCIZ  <CRLF>'AFTER 4TH CYCLE READ  '
070460  020122  052064  020110
070466  054503  046103  020105
070474  042522  042101  020040
070502     000

070503     200  043101  042524  MTC120: .ASCIZ  <CRLF>'AFTER 6TH CYCLE READ  '
070510  020122  052066  020110
070516  054503  046103  020105
070524  042522  042101  020040
070532     000
070533     200  043101  042524  MTD120: .ASCIZ  <CRLF>'AFTER 8TH CYCLE READ  '
070540  020122  052070  020110
070546  054503  046103  020105
070554  042522  042101  020040
070562     000

070563     200  043101  042524  MTE120: .ASCIZ  <CRLF>'AFTER 10TH CYCLE READ '
070570  020122  030061  044124
```

```
070576  041440  041531  042514
070604  051040  040505  020104
070612     000

070613     200  043101  042524   MTF120: .ASCIZ  <CRLF>'AFTER 12TH CYCLE READ '
070620  020122  031061  044124
070626  041440  041531  042514
070634  051040  040505  020104
070642     000

070643     106  047522  020115   MTG120: .ASCIZ  'FROM THE HIT/MISS REG. EXPECTED '
070650  044124  020105  044510
070656  027524  044515  051523
070664  051040  043505  020056
070672  054105  042520  052103
070700  042105  000040

070704  052200  042510  050040   MTA124: .ASCII  <CRLF>'THE PATTERN BEING USED IN THE MAINTENANCE '
070712  052101  042524  047122
070720  041040  044505  043516
070726  052440  042523  020104
070734  047111  052040  042510
070742  046440  044501  052116
070750  047105  047101  042503
070756     040
070757     122  043505  051511           .ASCIZ  'REGISTER WAS: '
070764  042524  020122  040527
070772  035123  000040

070776  051200  043105  051105   MTA126: .ASCIZ  <CRLF>'REFERENCED ADDRESS:'<TAB>
071004  047105  042503  020104
071012  042101  051104  051505
071020  035123  000011

071024  040600  051122  051117   MTB126: .ASCIZ  <CRLF>'ARROR ADDRESS REGISTER:'<TAB>
071032  040440  042104  042522
071040  051523  051040  043505
071046  051511  042524  035122
071054  000011

071056  050200  052101  042524   MTA131: .ASCIZ  <CRLF>'PATTERN BEING USED IN THE MAINTENANCE REGISTER:'<TAB>
071064  047122  041040  044505
071072  043516  052440  042523
071100  020104  047111  052040
071106  042510  046440  044501
071114  052116  047105  047101
071122  042503  051040  043505
071130  051511  042524  035122
071136  000011

071140  042600  050130  041505   MTB131: .ASCIZ  <CRLF>'EXPECTED ERROR REGISTER:'<TAB>
071146  042524  020104  051105
071154  047522  020122  042522
071162  044507  052123  051105
071170  004472     000

```

```
 11997  0/1173     200  047507  020124  MTC131: .ASCIZ  <CRLF>'GOT ERROR REGISTER:'<TAB>
 11998  071200  051105  047522  020122
 11999  071206  042522  044507  052123
 12000  071214  051105  004472     000
 12001
 12002  071221     200  051105  047522  MTA134: .ASCIZ  <CRLF>'ERROR ADR REG.'<TAB>'ERROR REG.'<CRLF>
 12003  071226  020122  042101  020122
 '2004  071234  042522  027107  042411
 12005  071242  051122  051117  051040
 12006  071250  043505  100056     000
 12007
 12008  071255     200  054105  042520  MTA135: .ASCIZ  <CRLF>'EXPECTED ERROR REG.:  '
 12009  071262  052103  042105  042440
 12010  071270  051122  051117  051040
 12011  071276  043505  035056  020040
 12012  071304     000
 12013
 1201/  071305     107  052117  042440  MTB135: .ASCIZ  'GOT ERROR REG.:  '
 12015  071312  051122  051117  051040
 12016  071320  043505  035056  020040
 12017  071326     000
 12018
 12019  071327     200  054105  042520  MTC135: .ASCIZ  <CRLF>'EXPECTED ERROR ADR REG.:  '
 12020  071334  052103  042105  042440
 12021  071342  051122  051117  040440
 12022  071350  051104  051040  043505
 12023  071356  035056  020040     000
 12024
 12025  071363     107  052117  042440  MTD135: .ASCIZ  'GOT ERROR ADR REG.:  '
 12026  071370  051122  051117  040440
 12027  071376  051104  051040  043505
 12028  071404  035056  020040     000
 12029
 12030
 12031  071411     015  053412  051101  MS01:   .ASCII  <15><12>/WARNING- THE SIZE OF MEMORY IS DIFFERENT THEN THAT/<CRLF>
 12032  071416  044516  043516  020055
 12033  071424  044124  020105  044523
 12034  071432  042532  047440  020106
 12035  071440  042515  047515  054522
 12036  071446  044440  020123  044504
 12037  071454  043106  051105  047105
 12038  071462  020124  044124  047105
 12039  071470  052040  040510  100124
 12040  071476  044440  042116  041511          .ASCIZ  / INDICATED BY THE SYSTEM SIZE REGISTER./
 12041  071504  052101  042105  041040
 12042  071512  020131  044124  020105
 12043  071520  054523  052123  046505
 12044  071526  051440  055111  020105
 12045  071534  042522  044507  052123
 12046  071542  051105  000056
 12047  071546  005015  044523  042532  MS02:   .ASCIZ  <15><12>/SIZE REG.     ACTUAL/
 12048  071554  051040  043505  020056
 12049  071562  020040  020040  041501
 12050  071570  052524  046101     000
 12051  071575     040  020040  020040  MS03:   .ASCIZ  /       /
 12052  071602  020040     000
```

```
12053  071605     200  050103  020125  MSG1:    .ASCIZ<CRLF>     "CPU UNDER TEST FOUND TO BE A "
12054  071612  047125  042504  020122
12055  071620  042524  052123  043040
12056  071626  052517  042116  052040
12057  071634  020117  042502  040440
12058  071642  000040
12059  071644  041113  030461  042455  MSG2:    .ASCIZ  'KB11-EM'<CRLF>
12060  071652  100115     000
12061  071655     113  030502  026461  MSG3:    .ASCIZ  'KB11-B/C'<CRLF>
12062  071662  027502  100103     000
12063  071667     113  030502  026461  MSG4:    .ASCIZ  'KB11-CM              '<CRLF>
12064  071674  046503  020040  020040
12065  071702  020040  020040  020040
12066  071710  020040  020040  020040
12067  071716  000200
12068  071720  041113  030461  042455  MSG5:    .ASCIZ  'KB11-E'<CRLF>
12069  071726  000200
12070  071730  005015  047516  046440  EM724:   .ASCIZ  <CR><LF>/NO MAP REGISTERS AVAILABLE FOR UNIBUS PARITY ERROR TEST/
12071  071736  050101  051040  043505
12072  071744  051511  042524  051522
12073  071752  040440  040526  046111
12074  071760  041101  042514  043040
12075  071766  051117  052440  044516
12076  071774  052502  020123  040520
12077  072002  044522  054524  042440
12078  072010  051122  051117  052040
12079  072016  051505  000124
12080
12081                                  ;THESE ARE THE ERROR MESSAGES:
12082
12083  072022  020101  042522  042506  EM1:     .ASCIZ  'A REFERENCE WHICH SHOULD HAVE BEEN A HIT WAS A MISS.'
12084  072030  042522  041516  020105
12085  072036  044127  041511  020110
12086  072044  044123  052517  042114
12087  072052  044040  053101  020105
12088  072060  042502  047105  040440
12089  072066  044040  052111  053440
12090  072074  051501  040440  046440
12091  072102  051511  027123     000
12092
12093  072107     125  042516  050130  EM2:     .ASCII  'UNEXPECTED ERROR DURING WORST CASE NOISE TEST ON '
12094  072114  041505  042524  020104
12095  072122  051105  047522  020122
12096  072130  052504  044522  043516
12097  072136  053440  051117  052123
12098  072144  041440  051501  020105
12099  072152  047516  051511  020105
12100  072160  042524  052123  047440
12101  072166  020116
12102  072170  040503  044103  020105           .ASCII  'CACHE DATA MEMORY.'<CRLF>
12103  072176  040504  040524  046440
12104  072204  046505  051117  027131
12105  072212     200
12106  072213     101  047040  047117           .ASCIZ  'A NON-CACHE DATA PARITY ERROR OCCURRED WHILE TESTING.'
12107  072220  041455  041501  042510
12108  072226  042040  052101  020101
```

```
072234  040520  044522  054524
072242  042440  051122  051117
072250  047440  041503  051125
072256  042522  020104  044127
072264  046111  020105  042524
072272  052123  047111  027107
072300     000

072301     127  051117  052123  EM3:    .ASCII  'WORST CASE NOISE TEST OF THE CACHE DATA MEMORY '
072306  041440  051501  020105
072314  047516  051511  020105
072322  042524  052123  047440
072330  020106  044124  020105
072336  040503  044103  020105
072344  040504  040524  046440
072352  046505  051117  020131
072360  043200  044501  042514          .ASCIZ  <CRLF>/FAILED WHILE GALLOPING 0'S./
072366  020104  044127  046111
072374  020105  040507  046114
072402  050117  047111  020107
072410  023460  027123     000

072415     127  051117  052123  EM4:    .ASCII  'WORST CASE NOISE TEST OF THE CACHE DATA MEMORY'
072422  041440  051501  020105
072430  047516  051511  020105
072436  042524  052123  047440
072444  020106  044124  020105
072452  040503  044103  020105
072460  040504  040524  046440
072466  046505  051117     131
072473     200  040506  046111          .ASCIZ  <CRLF>/FAILED WHILE GALLOPING 1'S./
072500  042105  053440  044510
072506  042514  043440  046101
072514  047514  044520  043516
072522  030440  051447  000056

072530  042103  054115  052040  EM5:    .ASCIZ  'CDMX TEST FAILURE.'<CRLF>'BAD CACHE GROUP 0 DATA READ.'
072536  051505  020124  040506
072544  046111  051125  027105
072552  041200  042101  041440
072560  041501  042510  043440
072566  047522  050125  030040
072574  042040  052101  020101
072602  042522  042101  000056

072610  042103  054115  052040  EM6:    .ASCIZ  'CDMX TEST FAILURE.'<CRLF>'BAD CACHE GROUP 1 DATA READ.'
072616  051505  020124  040506
072624  046111  051125  027105
072632  041200  042101  041440
072640  041501  042510  043440
072646  047522  050125  030440
072654  042040  052101  020101
072662  042522  042101  000056

072670  042103  054115  052040  EM7:    .ASCII  'CDMX TEST FAILURE.'<CRLF>'BAD MAIN MEMORY, EVEN WORD,'
072676  051505  020124  040506
```

```
072704  046111  051125  027105
072712  041200  042101  046440
072720  044501  020116  042515
072726  047515  054522  020054
072734  053105  047105  053440
072742  051117  026104
072746  042040  052101  020101          .ASCIZ  ' DATA READ.'
072754  042522  042101  000056

072762  042103  054115  052040  EM10:   .ASCII  'CDMX TEST FAILURE.'<CRLF>'BAD MAIN MEMORY, ODD WORD,'
072770  051505  020124  040506
072776  046111  051125  027105
073004  041200  042101  046440
073012  044501  020116  042515
073020  047515  054522  020054
073026  042117  020104  047527
073034  042122     054
073037     040  040504  040524          .ASCIZ  ' DATA READ.'
073044  051040  040505  027104
073052     000

073053     120  051101  052111  EM11:   .ASCIZ  'PARITY ERROR IN CACHE DATA MEMORY COUNT PATTERN TEST.'
073060  020131  051105  047522
073066  020122  047111  041440
073074  041501  042510  042040
073102  052101  020101  042515
073110  047515  054522  041440
073116  052517  052116  050040
073124  052101  042524  047122
073132  052040  051505  027124
073140     000

073141     102  042101  042040  EM12:   .ASCII  'BAD DATA WAS READ IN CACHE MEMORY COUNT PATTERN '
073146  052101  020101  040527
073154  020123  042522  042101
073162  044440  020116  040503
073170  044103  020105  042515
073176  047515  054522  041440
073204  052517  052116  050040
073212  052101  042524  047122
073220     040
073221     124  051505  027124          .ASCIZ  'TEST.'<CRLF>'BUT NO TRAP OR ABORT OCCURRED.'
073226  041200  052125  047040
073234  020117  051124  050101
073242  047440  020122  041101
073250  051117  020124  041517
073256  052503  051122  042105
073264  000056

073266  040503  044103  020105  EM13:   .ASCII  'CACHE MEMORY COUNT PATTERN TEST.'<CRLF>
073274  042515  047515  054522
073302  041440  052517  052116
073310  050040  052101  042524
073316  047122  052040  051505
073324  027124     200
073327     105  051122  051117          .ASCIZ  'ERROR SUMMARY.'
```

```
12221  073334  051440  046525  040515
12222  073342  054522  000056
12223
12224  073346  052600  042516  050130  EM14:   .ASCIZ  <CRLF>'UNEXPECTED PARITY ERROR TRAP.'
12225  073354  041505  042524  020104
12226  073362  040520  044522  054524
12227  073370  042440  051122  051117
12228  073376  052040  040522  027120
12229  073404     000
12230
12231  073405     052  025052  042524  EM15:   .ASCIZ  '***TEST ABORTED! GOING TO NEXT TEST.***'
12232  073412  052123  040440  047502
12233  073420  052122  042105  020041
12234  073426  047507  047111  020107
12235  073434  047524  047040  054105
12236  073442  020124  042524  052123
12237  073450  025056  025052     000
12238
12239
12240  073455     103  041501  042510  EM16:   .ASCIZ  'CACHE DATA MEMORY DUAL ADDRESS TEST FAILED.'
12241  073462  042040  052101  020101
12242  073470  042515  047515  054522
12243  073476  042040  040525  020114
12244  073504  042101  051104  051505
12245  073512  020123  042524  052123
12246  073520  043040  044501  042514
12247  073526  027104     000
12248
12249  073531     103  041501  042510  EM17:   .ASCIZ  'CACHE DATA MEMORY BYTE ENABLE LOGIC TEST FAILED.'
12250  073536  042040  052101  020101
12251  073544  042515  047515  054522
12252  073552  041040  052131  020105
12253  073560  047105  041101  042514
12254  073566  046040  043517  041511
12255  073574  052040  051505  020124
12256  073602  040506  046111  042105
12257  073610  000056
12258
12259          073531                  EM20=EM17
12260
12261  073612  040503  044103  020105  EM21:   .ASCIZ  'CACHE DATA MEMORY CHIP SELECTION LOGIC TEST FAILED.'
12262  073620  040504  040524  046440
12263  073626  046505  051117  020131
12264  073634  044103  050111  051440
12265  073642  046105  041505  044524
12266  073650  047117  046040  043517
12267  073656  041511  052040  051505
12268  073664  020124  040506  046111
12269  073672  042105  000056
12270
12271  073676  042101  051104  051505  EM22:   .ASCII  'ADDRESS MULTIPLEXER TEST WAS UNABLE TO FORCE'
12272  073704  020123  052515  052114
12273  073712  050111  042514  042530
12274  073720  020122  042524  052123
12275  073726  053440  051501  052440
12276  073734  040516  046102  020105
```

```
073742  047524  043040  051117
073750  042503
073752  040440  050040  051101                .ASCII  ' A PARITY ERROR, USING THE '<CRLF>
073760  052111  020131  051105
073766  047522  026122  052440
073774  044523  043516  052040
074002  042510  100040
074006  040515  047111  042524                .ASCII  'MAINTENANCE REGISTER, ON THE'
074014  040516  041516  020105
074022  042522  044507  052123
074030  0511U5  020054  047117
074036  052040  042510
074042  046440  044501  020116                .ASCIZ  ' MAIN MEMORY ADDRESS AND CONTROL LINES.'
074050  042515  047515  054522
074056  040440  042104  042522
074064  051523  040440  042116
074072  041440  047117  051124
074100  046117  046040  047111
074106  051505  000056

074112  042101  051104  051505  EM23:   .ASCII  'ADDRESS MULTIPLEXER, AMX, CPU INPUTS TEST FAILED.'
074120  020123  052515  052114
074126  050111  042514  042530
074134  026122  040440  054115
074142  020054  050103  020125
074150  047111  052520  051524
074156  052040  051505  020124
074164  040506  046111  042105
074172     056
074173     200  051105  047522          .ASCIZ  <CRLF>'ERROR ADDRESS REGISTER NOT SET CORRECTLY.'
074200  020122  042101  051104
074206  051505  020123  042522
074214  044507  052123  051105
074222  047040  052117  051440
074230  052105  041440  051117
074236  042522  052103  054514
074244  000056

        073676                  EM24=EM22

        074112                  EM25=EM23

074246  042101  051104  051505  EM26:   .ASCII  'ADDRESS MEMORY, ADDRESS COMPARATOR TEST FAILURE.'
074254  020123  042515  047515
074262  054522  020054  042101
074270  051104  051505  020123
074276  047503  050115  051101
074304  052101  051117  052040
074312  051505  020124  040506
074320  046111  051125  027105
074326  040600  020116  042101          .ASCII  <CRLF>'AN ADDRESS WHICH SHOULD HAVE BEEN A HIT WAS'
074334  051104  051505  020123
074342  044127  041511  020110
074350  044123  052517  042114
074356  044040  053101  020105
074364  042502  047105  040440
```

```
074372  044040  052111  053440
074400  051501
074402  040440  046440  051511                  .ASCIZ  ' A MISS.'
074410  027123     000

074413     101  042104  042522  EM27:   .ASCII  'ADDRESS MEMORY, ADDRESS COMPARATOR TEST FAILURE.'
074420  051523  046440  046505
074426  051117  026131  040440
074434  042104  042522  051523
074442  041440  046517  040520
074450  040522  047524  020122
074456  042524  052123  043040
074464  044501  052514  042522
074472     056
074473     200  047101  040440          .ASCII  <CRLF>'AN ADDRESS WHICH SHOULD HAVE BEEN A MISS '
074500  042104  042522  051523
074506  053440  044510  044103
074514  051440  047510  046125
074522  020104  040510  042526
074530  041040  042505  020116
074536  020101  044515  051523
074544     040
074545     127  051501  040440          .ASCIZ  'WAS A HIT.'
074552  044040  052111  000056

        073676                  EM30-EM22

074560  042101  051104  051505  EM31:   .ASCII  'ADDRESS MULTIPLEXER, AMX, UNIBUS INPUTS TEST FAILED.'
074566  020123  052515  052114
074574  050111  042514  042530
074602  026122  040440  054115
074610  020054  047125  041111
074616  051525  044440  050116
074624  052125  020123  042524
074632  052123  043040  044501
074640  042514  027104
074644  042600  051122  051117          .ASCIZ  <CRLF>'ERROR ADDRESS REGISTER NOT SET CORRECTLY.'
074652  040440  042104  042522
074660  051523  051040  043505
074666  051511  042524  020122
074674  047516  020124  042523
074702  020124  047503  051122
074710  041505  046124  027131
074716     000

        073676                  EM32-EM22

        074560                  EM33=EM31

074717     101  042104  042522  EM34:   .ASCII  'ADDRESS MULTIPLEXER, AMX, DUAL ADDRESS TEST,'<CRLF>
074724  051523  046440  046125
074732  044524  046120  054105
074740  051105  020054  046501
074746  026130  042040  040525
074754  020114  042101  051104
074762  051505  020123  042524
```

```
12389 074770  052123  100054
12390 074774  047117  041440  052520          .ASCIZ  'ON CPU INPUTS, FAILED.'
12391 075002  044440  050116  052125
12392 075010  026123  043040  044501
12393 075016  042514  027104     000
12394
12395 075023     101  042104  042522  EM35:   .ASCII  'ADDRESS MULTIPLEXER, AMX, DUAL ADDRESS TEST,'<CRLF>
12396 075030  051523  046440  046125
12397 075036  044524  046120  054105
12398 075044  051105  020054  046501
12399 075052  026130  042040  040525
12400 075060  020114  042101  051104
12401 075066  051505  020123  042524
12402 075074  052123  100054
12403 075100  047117  052440  044516          .ASCIZ  'ON UNIBUS INPUTS, FAILED.'
12404 075106  052502  020123  047111
12405 075114  052520  051524  020054
12406 075122  040506  046111  042105
12407 075130  000056
12408
12409 075132  042101  051104  051505  EM36:   .ASCII  'ADDRESS MEMORY COUNT PATTERN TEST FAILURE,'<CRLF>
12410 075140  020123  042515  047515
12411 075146  054522  041440  052517
12412 075154  052116  050040  052101
12413 075162  042524  047122  052040
12414 075170  051505  020124  040506
12415 075176  046111  051125  026105
12416 075204     200
12417 075205     116  020117  040520          .ASCIZ  'NO PARITY ERROR OCCURS, BUT CAN NOT GET A HIT.'
12418 075212  044522  054524  042440
12419 075220  051122  051117  047440
12420 075226  041503  051125  026123
12421 075234  041040  052125  041440
12422 075242  047101  047040  052117
12423 075250  043440  052105  040440
12424 075256  044040  052111  000056
12425
12426 075264  042101  051104  051505  EM37:   .ASCIZ  'ADDRESS MEMORY COUNT PATTERN TEST, ERROR SUMMARY.'
12427 075272  020123  042515  047515
12428 075300  054522  041440  052517
12429 075306  052116  050040  052101
12430 075314  042524  047122  052040
12431 075322  051505  026124  042440
12432 075330  051122  051117  051440
12433 075336  046525  040515  054522
12434 075344  000056
12435
12436 075346  042101  051104  051505  EM40:   .ASCII  'ADDRESS MEMORY COUNT PATTERN TEST FAILURE,'<CRLF>
12437 075354  020123  042515  047515
12438 075362  054522  041440  052517
12439 075370  052116  050040  052101
12440 075376  042524  047122  052040
12441 075404  051505  020124  040506
12442 075412  046111  051125  026105
12443 075420     200
12444 075421     103  041501  042510          .ASCII  'CACHE MEMORY ADDRESS PARITY ERROR OCCURRED'
```

```
075426  046440  046505  051117
075434  020131  042101  051104
075442  051505  020123  040520
075450  044522  054524  042440
075456  051122  051117  047440
075464  041503  051125  042522
075472     104
075473     040  052101  052040           .ASCIZ  ' AT THE TEST ADDRESS.'
075500  042510  052040  051505
075506  020124  042101  051104
075514  051505  027123     000

075521     101  042104  042522   EM41:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILED TO GET '
075526  051523  046440  046505
075534  051117  020131  052504
075542  046101  040440  042104
075550  042522  051523  052040
075556  051505  020124  040506
075564  046111  042105  052040
075572  020117  042507  020124
075600  020101  044510  020124           .ASCII  'A HIT AT A TEST ADDRESS.'<CRLF>
075606  052101  040440  052040
075614  051505  020124  042101
075622  051104  051505  026123
075630     200
075631     127  044510  042514           .ASCIZ  'WHILE WRITING THE ADDRESS MEMORY LOCATIONS.'
075636  053440  044522  044524
075644  043516  052040  042510
075652  040440  042104  042522
075660  051523  046440  046505
075666  051117  020131  047514
075674  040503  044524  047117
075702  027123     000

075705     101  042104  042522   EM42:   .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILED TO GET'
075712  051523  046440  046505
075720  051117  020131  052504
075726  046101  040440  042104
075734  042522  051523  052040
075742  051505  020124  040506
075750  046111  042105  052040
075756  020117  042507     124
075763     101  044040  052111           .ASCII  'A HIT AT A TEST ADDRESS.'<CRLF>
075770  040440  020124  020101
075776  042524  052123  040440
076004  042104  042522  051523
076012  100054

076014  044127  046111  020105           .ASCII  'WHILE READING BACK THE ADDRESS MEMORY LOCATIONS.'<CRLF><LF>
076022  042522  042101  047111
076030  020107  040502  045503
076036  052040  042510  040440
076044  042104  042522  051523
076052  046440  046505  051117
076060  020131  047514  040503
076066  044524  047117  027123
```

```
12501 076074 005200
12502 076076 052133 044510 020123         .ASCIZ  '[THIS PROBLEM MIGHT BE CORRECTED BY ECO M8182-4]'<CRLF>
12503 076104 051120 041117 042514
12504 076112 020115 044515 044107
12505 076120 020124 042502 041440
12506 076126 051117 042522 052103
12507 076134 042105 041040 020131
12508 076142 041505 020117 034115
12509 076150 034061 026462 056464
12510 076156 000200
12511
12512
12513
12514 076160 042101 051104 051505  EM43:  .ASCII  'ADDRESS MEMORY DUAL ADDRESS TEST FAILURE,'<CRLF>
12515 076166 020123 042515 047515
12516 076174 054522 042040 040525
12517 076202 020114 042101 051104
12518 076210 051505 020123 042524
12519 076216 052123 043040 044501
12520 076224 052514 042522 100054
12521 076232 040503 044103 020105         .ASCIZ  'CACHE ADDRESS MEMORY PARITY ERROR OCCURRED.'
12522 076240 042101 051104 051505
12523 076246 020123 042515 047515
12524 076254 054522 050040 051101
12525 076262 052111 020131 051105
12526 076270 047522 020122 041517
12527 076276 052503 051122 042105
12528 076304 000056
12529
12530
12531 076306 040515 047111 046440  EM44:  .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED,'
12532 076314 046505 051117 020131
12533 076322 054502 042524 046440
12534 076330 051501 020113 042507
12535 076336 042516 040522 047524
12536 076344 020122 042524 052123
12537 076352 043040 044501 042514
12538 076360 026104
12539 076362 042040 044517 043516         .ASCII  ' DOING CPU DATOB.'<CRLF>
12540 076370 041440 052520 042040
12541 076376 052101 041117 100056
12542 076404 020101 040515 047111         .ASCII  'A MAIN MEMORY ADDRESS AND CONTROL LINE '
12543 076412 046440 046505 051117
12544 076420 020131 042101 051104
12545 076426 051505 020123 047101
12546 076434 020104 047503 052116
12547 076442 047522 020114 044514
12548 076450 042516    040
12549 076453    120 051101 052111         .ASCIZ  'PARITY ERROR OCCURRED.'
12550 076460 020131 051105 047522
12551 076466 020122 041517 052503
12552 076474 051122 042105 000056
12553
12554 076502 040515 047111 046440  EM45:  .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED,'
12555 076510 046505 051117 020131
12556 076516 054502 042524 046440
```

```
12557  076524  051501  020113  042507
12558  076532  042516  040522  047524
12559  076540  020122  042524  052123
12560  076546  043040  044501  042514
12561  076554  026104
12562  076556  042040  044517  043516          .ASCII  ' DOING UNIBUS DATOB.'<CRLF>
12563  076564  052440  044516  052502
12564  076572  020123  040504  047524
12565  076600  027102     200
12566  076603     101  046440  044501          .ASCII  'A MAIN MEMORY ADDRESS AND CONTROL LINE '
12567  076610  020116  042515  047515
12568  076616  054522  040440  042104
12569  076624  042522  051523  040440
12570  076632  042116  041440  047117
12571  076640  051124  046117  04604C
12572  076646  047111  020105
12573  076652  040520  044522  054524          .ASCIZ  'PARITY ERROR OCCURRED.'
12574  076660  042440  051122  051117
12575  076666  047440  041503  051125
12576  076674  042522  027104     000
12577
12578  076701     115  044501  020116  EM46:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED.'
12579  076706  042515  047515  054522
12580  076714  041040  052131  020105
12581  076722  040515  045523  043440
12582  076730  047105  051105  052101
12583  076736  051117  052040  051505
12584  076744  020124  040506  046111
12585  076752  042105     056
12586  076755     200  051127  047117          .ASCIZ  <CRLF>'WRONG BYTE WRITTEN, ON A CPU DATOB.'
12587  076762  020107  054502  042524
12588  076770  053440  044522  052124
12589  076776  047105  020054  047117
12590  077004  040440  041440  052520
12591  077012  042040  052101  041117
12592  077020  000056
12593
12594  077022  040515  047111  046440  EM47:   .ASCII  'MAIN MEMORY BYTE MASK GENERATOR TEST FAILED.'
12595  077030  046505  051117  020131
12596  077036  054502  042524  046440
12597  077044  051501  020113  042507
12598  077052  042516  040522  047524
12599  077060  020122  042524  052123
12600  077066  043040  044501  042514
12601  077074  027104
12602  077076  053600  047522  043516          .ASCIZ  <CRLF>'WRONG BYTE WRITTEN, ON A UNIBUS DATOB.'
12603  077104  041040  052131  020105
12604  077112  051127  052111  042524
12605  077120  026116  047440  020116
12606  077'26  020101  047125  041111
12607  077134  051525  042040  052101
12608  077142  041117  000056
12609
12610          076306                  EM50-EM44
12611
12612          076502                  EM51 EM45
```

```
12613
12614            076701                      EM52=EM46
12615
12616            077022                      EM53=EM47
12617
12618  077146    040503  044103  020105      EM54:   .ASCII  'CACHE ADDRESS MEMORY POWER UP INVALIDATOR TEST FAILED.'
12619  077154    042101  051104  051505
12620  077162    020123  042515  047515
12621  077170    054522  050040  053517
12622  077176    051105  052440  020120
12623  077204    047111  040526  044514
12624  077212    040504  047524  020122
12625  077220    042524  052123  043040
12626  077226    044501  042514  027104
12627  077234    041600  041501  042510              .ASCII  <CRLF>'CACHE DATA OR ADDRESS MEMORY PARITY '
12628  077242    042040  052101  020101
12629  077250    051117  040440  042104
12630  077256    042522  051523  046440
12631  077264    046505  051117  020131
12632  077272    040520  044522  054524
12633  077300       040
12634  077301       105  051122  051117              .ASCIZ  'ERROR DETECTED.'
12635  077306    042040  052105  041505
12636  077314    042524  027104     000
12637  077321       103  051103  041440      EM55:   .ASCIZ  /CCR COULD NOT BE CLEARED/
12638  077326    052517  042114  047040
12639  077334    052117  041040  020105
12640  077342    046103  040505  042522
12641  077350    000104
12642  077352    053111  051523  024040      EM56:   .ASCIZ  /IVSS (BIT 14) COULD NOT BE SET IN CCR/
12643  077360    044502  020124  032061
12644  077366    020051  047503  046125
12645  077374    020104  047516  020124
12646  077402    042502  051440  052105
12647  077410    044440  020116  041503
12648  077416    000122
12649  077420    053111  051523  041440      EM57:   .ASCIZ  /IVSS COULD NOT BE CLEARED IN CCR/
12650  077426    052517  042114  047040
12651  077434    052117  041040  020105
12652  077442    046103  040505  042522
12653  077450    020104  047111  041440
12654  077456    051103     000
12655  077461       126  044523  020125      EM60:   .ASCIZ  /VSIU (BIT 13) COULD NOT BE SET/
12656  077466    041050  052111  030440
12657  077474    024463  041440  052517
12658  077502    042114  047040  052117
12659  077510    041040  020105  042523
12660  077516    000124
12661  077520    041526  050111  042040      EM61:   .ASCIZ  /VCIP DID NOT CLEAR AFTER CACHE FLUSH (ON SETTING VSIU)/
12662  077526    042111  047040  052117
12663  077534    041440  042514  051101
12664  077542    040440  052106  051105
12665  077550    041440  041501  042510
12666  077556    043040  052514  044123
12667  077564    024040  047117  051440
12668  077572    052105  044524  043516
```

```
077600  053040  044523  024525
077606     000
077607     126  044523  020125    EM62:   .ASCIZ  /VSIU COULD NOT BE CLEARED/
077614  047503  046125  020104
077622  047516  020124  042502
077630  041440  042514  051101
077636  042105     000
077641     126  044503  020120    EM63:   .ASCIZ  /VCIP DID NOT SET WHEN CACHE FLUSH BIT WAS SET/
077646  044504  020104  047516
077654  020124  042523  020124
077662  044127  047105  041440
077670  041501  042510  043040
077676  052514  044123  041040
077704  052111  053440  051501
077712  051440  052105     000
077717     126  044523  020125    EM64:   .ASCIZ  /VSIU DID NOT SWITCH WHEN CACHE FLUSH BIT WAS SFT/
077724  044504  020104  047516
077732  020124  053523  052111
077740  044103  053440  042510
077746  020116  040503  044103
077754  020105  046106  051525
077762  020110  044502  020124
077770  040527  020123  042523
077776  000124
100000  051526  052511  051440    EM65:   .ASCIZ  /VSIU SWITCHED WHEN CACHE FLUSH WAS DONE,WITH IVSS SET/
100006  044527  041524  042510
100014  020104  044127  047105
100022  041440  041501  042510
100030  043040  052514  044123
100036  053440  051501  042040
100044  047117  026105  044527
100052  044124  044440  051526
100060  020123  042523  000124
100066  042524  052123  042055    EM66:   .ASCII  /TEST-DATA REFERENCE NOT A MISS/
100074  052101  020101  042522
100102  042506  042522  041516
100110  020105  047516  020124
100116  020101  044515  051523
100124  005015  040526  044514            .ASCIZ  <15><12>/VALID STORE NOT SWITCHED ON CACHE FLUSH/
100132  020104  052123  051117
100140  020105  047516  020124
100146  053523  052111  044103
100154  042105  047440  020116
100162  040503  044103  020105
100170  046106  051525  000110
100176  042524  052123  042055    EM67:   .ASCII  /TEST-DATA REFERENCE NOT A MISS/
100204  052101  020101  042522
100212  042506  042522  041516
100220  020105  047516  020124
100226  020101  044515  051523
100234  005015  040526  044514            .ASCIZ  <15><12>/VALID STORE NOT INVALIDATED ON CACHE FLUSH/
100242  020104  052123  051117
100250  020105  047516  020124
100256  047111  040526  044514
100264  040504  042524  020104
100272  047117  041440  041501
```

```
100300  042510  043040  052514
100306  044123     000
100311     124  051505  026524   EM70:  .ASCII  /TEST-DATA REFERENCE NOT A HIT/
100316  040504  040524  051040
100324  043105  051105  047105
100332  042503  047040  052117
100340  040440  044040  052111
100346  005015  051106  046517          .ASCIZ  <15><12>/FROM THE GROUP AND VALID STORE BEING CHECKED/
100354  052040  042510  043440
100362  047522  050125  040440
100370  042116  053040  046101
100376  042111  051440  047524
100404  042522  041040  044505
100412  043516  041440  042510
100420  045503  042105     000
100425     104  052101  020101   EM71:  .ASCIZ  /DATA ERROR ON READING CACHED LOCATION/
100432  051105  047522  020122
100440  047117  051040  040505
100446  044504  043516  041440
100454  041501  042510  020104
100462  047514  040503  044524
100470  047117     000
100473     124  051505  026524   EM72:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
100500  040504  040524  051040
100506  043105  051105  047105
100514  042503  047040  052117
100522  040440  046440  051511
100530     123
100531     015  041412  041501          .ASCIZ  <15><12>/CACHE DOES NOT TURN OFF, WHEN FLUSH DONE WITH IVSS SET/
100536  042510  042040  042517
100544  020123  047516  020124
100552  052524  047122  047440
100560  043106  020054  044127
100566  047105  043040  052514
100574  044123  042040  047117
100602  020105  044527  044124
100610  044440  051526  020123
100616  042523  000124
100622  042524  052123  042055   EM73:  .ASCII  /TEST-DATA REFERENCE NOT A HIT/
100630  052101  020101  042522
100636  042506  042522  041516
100644  020105  047516  020124
100652  020101  044510     124
100657     015  041412  041501          .ASCIZ  <15><12>/CACHE DOES NOT TURN ON AFTER TURNING OFF/
100664  042510  042040  042517
100672  020123  047516  020124
100700  052524  047122  047440
100706  020116  043101  042524
100714  020122  052524  047122
100722  047111  020107  043117
100730  000106
100732  042524  052123  042055   EM74:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
100740  052101  020101  042522
100746  042506  042522  041516
100754  020105  047516  020124
100762  020101  044515  051523
```

```
 12781  100770  005015  040503  044103          .ASCIZ  <15><12>/CACHE BYPASS DID NOT FORCE A MISS/
 12782  100776  020105  054502  040520
 12783  101004  051523  042040  042111
 12784  101012  047040  052117  043040
 12785  101020  051117  042503  040440
 12786  101026  046440  051511  000123
 12787  101034  042524  052123  042055  EM75:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
 12788  101042  052101  020101  042522
 12789  101050  042506  042522  041516
 12790  101056  020105  047516  020124
 12791  101064  020101  044515  051523
 12792  101072  005015  040503  044103          .ASCIZ  <15><12>/CACHE BYPASS DID NOT INVALIDATE CACHED DATA/
 12793  101100  020105  054502  040520
 12794  101106  051523  042040  042111
 12795  101114  047040  052117  044440
 12796  101122  053116  046101  042111
 12797  101130  052101  020105  040503
 12798  101136  044103  042105  042040
 12799  101144  052101  000101
 12800  101150  042524  052123  042055  EM76:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
 12801  101156  052101  020101  042522
 12802  101164  042506  042522  041516
 12803  101172  020105  047516  020124
 12804  101200  020101  044515  051523
 12805  101206  005015  051501  041122          .ASCIZ  <15><12>/ASRB DID NOT FORCE A MISS ON THE OPERAND/
 12806  101214  042040  042111  047040
 12807  101222  052117  043040  051117
 12808  101230  042503  040440  046440
 12809  101236  051511  020123  047117
 12810  101244  052040  042510  047440
 12811  101252  042520  040522  042116
 12812  101260     000
 12813  101261     124  051505  026524  EM77:  .ASCII  /TEST-DATA REFERENCE NOT A MISS/
 12814  101266  040504  040524  051040
 12815  101274  043105  051105  047105
 12816  101302  042503  047040  052117
 12817  101310  040440  046440  051511
 12818  101316     123
 12819  101317     015  041412  041501          .ASCII  <15><12>/CACHED OPERAND NOT INVALIDATED ON ASRB EXECUTION/<CRLF>
 12820  101324  042510  020104  050117
 12821  101332  051105  047101  020104
 12822  101340  047516  020124  047111
 12823  101346  040526  044514  040504
 12824  101354  042524  020104  047117
 12825  101362  040440  051123  020102
 12826  101370  054105  041505  052125
 12827  101376  047511  100116
 12828  101402  052133  044510  020123          .ASCIZ  /[THIS PROBLEM MIGHT BE CORRECTED BY ECO M8182-4]/<CRLF>
 12829  101410  051120  041117  042514
 12830  101416  020115  044515  044107
 12831  101424  020124  042502  041440
 12832  101432  051117  042522  052103
 12833  101440  042105  041040  020131
 12834  101446  041505  020117  034115
 12835  101454  034061  026462  056464
 12836  101462  000200
```

```
12837  101464  042524  052123  042055   EM100: .ASCIZ  /TEST-DATA COULD NOT BE MADE HIT/
12838  101472  052101  020101  047503
12839  101500  046125  020104  047516
12840  101506  020124  042502  046440
12841  101514  042101  020105  044510
12842  101522  000124
12843  101524  047516  050040  051101   EM103: .ASCIZ  /NO PARITY ERROR TRAP ON VALID STORE PARITY ERROR/
12844  101532  052111  020131  051105
12845  101540  047522  020122  051124
12846  101546  050101  047440  020116
12847  101554  040526  044514  020104
12848  101562  052123  051117  020105
12849  101570  040520  044522  054524
12850  101576  042440  051122  051117
12851  101604     000
12852  101605     124  051505  026524   EM104: .ASCII  /TEST-DATA-REFERENCE GIVING VALID STORE PARITY/
12853  101612  040504  040524  051055
12854  101620  043105  051105  047105
12855  101626  042503  043440  053111
12856  101634  047111  020107  040526
12857  101642  044514  020104  052123
12858  101650  051117  020105  040520
12859  101656  044522  054524
12860  101662  005015  051105  047522          .ASCIZ  <15><12>/ERROR WAS NOT A MISS/
12861  101670  020122  040527  020123
12862  101676  047516  020124  020101
12863  101704  044515  051523     000
12864  101711     106  050126  020105   EM105: .ASCIZ  /FVPE DID NOT GET CLEARED AFTER VSPE OCCURED/
12865  101716  044504  020104  047516
12866  101724  020124  042507  020124
12867  101732  046103  040505  042522
12868  101740  020104  043101  042524
12869  101746  020122  051526  042520
12870  101754  047440  041503  051125
12871  101762  042105     000
12872  101765     126  046101  042111   EM106: .ASCIZ  /VALID-STORE-PARITY-ERROR BIT DID NOT SET IN CCR ON VSPE/
12873  101772  051455  047524  042522
12874  102000  050055  051101  052111
12875  102006  026531  051105  047522
12876  102014  020122  044502  020124
12877  102022  044504  020104  047516
12878  102030  020124  042523  020124
12879  102036  047111  041440  051103
12880  102044  047440  020116  051526
12881  102052  042520     000
12882  102055     106  051501  020124   EM107: .ASCII  /FAST ADDRESS MEMORY PARITY ERROR BITS (4,5) NOT/
12883  102062  042101  051104  051505
12884  102070  020123  042515  047515
12885  102076  054522  050040  051101
12886  102104  052111  020131  051105
12887  102112  047522  020122  044502
12888  102120  051524  024040  026064
12889  102126  024465  047040  052117
12890  102134  005015  042523  020124          .ASCIZ  <15><12>/SET CORRECTLY IN MSER ON VSPE/
12891  102142  047503  051122  041505
12892  102150  046124  020131  047111
```

```
12893  102156  046440  042523  020122
12894  102164  047117  053040  050123
12895  102172  000105
12896  102174  051526  052511  051440   EM110: .ASCIZ  /VSIU SWITCHED ON VSPE/
12897  102202  044527  041524  042510
12898  102210  020104  047117  053040
12899  102216  050123  000105
12900  102222  042515  047515  054522  EM111:  .ASCIZ  /MEMORY SYSTEM ERROR REGISTER COULD NOT BE CLEARED/
12901  102230  051440  051531  042524
12902  102236  020115  051105  047522
12903  102244  020122  042522  044507
12904  102252  052123  051105  041440
12905  102260  052517  042114  047040
12906  102266  052117  041040  020105
12907  102274  046103  040505  042522
12908  102302  000104
12909  102304  051526  042520  041440   EM112: .ASCIZ  /VSPE COULD NOT BE CLEARED IN CCR/
12910  102312  052517  042114  047040
12911  102320  052117  041040  020105
12912  102326  046103  040505  042522
12913  102334  020104  047111  041440
12914  102342  051103     000
12915  102345     124  051505  026524   EM113: .ASCIZ  /TEST-DATA-REFERENCE NOT A HIT/
12916  102352  040504  040524  051055
12917  102360  043105  051105  047105
12918  102366  042503  047040  052117
12919  102374  040440  044040  052111
12920  102402     000
12921  102403     124  051505  026524  EM115:  .ASCII  /TEST-DATA-REFERNECE NOT A MISS/
12922  102410  040504  040524  051055
12923  102416  043105  051105  042516
12924  102424  042503  047040  052117
12925  102432  040440  046440  051511
12926  102440     123
12927  102441     015  041412  041501          .ASCIZ  <15><12>/CACHE DID NOT TURN OFF ON BACK-TO-BACK FLUSH/
12928  102446  042510  042040  042111
12929  102454  047040  052117  052040
12930  102462  051125  020116  043117
12931  102470  020106  047117  041040
12932  102476  041501  026513  047524
12933  102504  041055  041501  020113
12934  102512  046106  051525  000110
12935
12936  102520  054502  020120  044502  EM123:  .ASCIZ  ?BYP BIT IN KIPDR COULD NOT BE CLEARED?
12937  102526  020124  047111  045440
12938  102534  050111  051104  041440
12939  102542  052517  042114  047040
12940  102550  052117  041040  020105
12941  102556  046103  040505  042522
12942  102564  000104
12943  102566  054502  020120  044502  EM124:  .ASCIZ  ?BYP BIT IN KIPDR COULD NOT BE SET?
12944  102574  020124  047111  045440
12945  102602  050111  051104  041440
12946  102610  052517  042114  047040
12947  102616  052117  041040  020105
12948  102624  042523  000124
```

```
102630 042524 052123 042055 EM125:  .ASCIZ  /TEST-DATA COULD NOT BE MADE HIT/
102636 052101 020101 047503
102644 046125 020104 047516
102652 020124 042502 046440
102660 042101 020105 044510
102666 000124
102670 042524 052123 042055  EM126: .ASCII  /TEST-DATA REFERENCE NOT A MISS/
102676 052101 020101 042522
102704 042506 042522 041516
102712 020105 047516 020124
102720 020101 044515 051523
102726 005015 040503 044103         .ASCIZ  <15><12>/CACHED DATA WAS NOT FORCED A MISS ON VIRTUAL PAGE BYPASS/
102734 042105 042040 052101
102742 020101 040527 020123
102750 047516 020124 047506
102756 041522 042105 040440
102764 046440 051511 020123
102772 047117 053040 051111
103000 052524 046101 050040
103006 043501 020105 054502
103014 040520 051523    000
103021    124 051505 020124  EM127: .ASCII  /TEST DATA REFERENCE NOT A MISS/
103026 040504 040524 051040
103034 043105 051105 047105
103042 042503 047040 052117
103050 040440 046440 051511
103056    123
103057    015 041412 041501         .ASCIZ  <15><12>/CACHED DATA WAS NOT INVALIDATED ON VIRTUAL PAGE BYPASS/
103064 042510 020104 040504
103072 040524 053440 051501
103100 047040 052117 044440
103106 053116 046101 042111
103114 052101 042105 047440
103122 020116 044526 052122
103130 040525 020114 040520
103136 042507 041040 050131
103144 051501 000123
103150 054502 020120 044502 EM130:  .ASCIZ  ?BYP BIT IN SIPDR COULD NOT BE CLEARED?
103156 020124 047111 051440
103164 050111 051104 041440
103172 052517 042114 047040
103200 052117 041040 020105
103206 046103 040505 042522
103214 000104
103216 054502 020120 044502 EM131:  .ASCIZ  ?BYP BIT IN SIPDR COULD NOT BE SET?
103224 020124 047111 051440
103232 050111 051104 041440
103240 052517 042114 047040
103246 052117 041040 020105
103254 042523 000124
103260 054502 020120 044502 EM132:  .ASCIZ  ?BYP BIT IN UIPDR COULD NOT BE CLEARED?
103266 020124 047111 052440
103274 050111 051104 041440
103302 052517 042114 047040
103310 052117 041040 020105
103316 046103 040505 042522
```

```
13005  103324  000104
13006  103326  054502  020120  044502  EM133:  .ASCIZ  ?BYP BIT IN UIPDR COULD NOT BE SET?
13007  103334  020124  047111  052440
13008  103342  050111  051104  041440
13009  103350  052517  042114  047040
13010  103356  052117  041040  020105
13011  103364  042523  000124
13012  103370  040503  044103  020105  EM136:  .ASCII  'CACHE ADDRESS MEMORY PARITY LOGIC TEST FAILED.'<CRLF>
13013  103376  042101  051104  051505
13014  103404  020123  042515  047515
13015  103412  054522  050040  051101
13016  103420  052111  020131  047514
13017  103426  044507  020103  042524
13018  103434  052123  043040  044501
13019  103442  042514  027104     200
13020  103447     125  040516  046102          .ASCII  'UNABLE TO FORCE A PARITY ERROR ON THE LOW BYTE '
13021  103454  020105  047524  043040
13022  103462  051117  042503  040440
13023  103470  050040  051101  052111
13024  103476  020131  051105  047522
13025  103504  020122  047117  052040
13026  103512  042510  046040  053517
13027  103520  041040  052131  020105
13028  103526  043117  040440  020116          .ASCIZ  'OF AN ADDRESS.'<CRLF>'USING THE MAINTENANCE REGISTER.'
13029  103534  042101  051104  051505
13030  103542  026123  052600  044523
13031  103550  043516  052040  042510
13032  103556  046440  044501  052116
13033  103564  047105  047101  042503
13034  103572  051040  043505  051511
13035  103600  042524  027122     000
13036
13037  103605     103  041501  042510  EM137:  .ASCII  'CACHE ADDRESS MEMORY PARITY LOGIC TEST FAILED.'
13038  103612  040440  042104  042522
13039  103620  051523  046440  046505
13040  103626  051117  020131  040520
13041  103634  044522  054524  046040
13042  103642  043517  041511  052040
13043  103650  051505  020124  040506
13044  103656  046111  042105     056
13045  103663     200  047125  041101          .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR ON THE HIGH BYTE '
13046  103670  042514  052040  020117
13047  103676  047506  041522  020105
13048  103704  020101  040520  044522
13049  103712  054524  042440  051122
13050  103720  051117  047440  020116
13051  103726  044124  020105  044510
13052  103734  044107  041040  052131
13053  103742  020105
13054  103744  043117  040440  020116          .ASCIZ  'OF AN ADDRESS.'<CRLF>'USING THE MAINTENANCE REGISTER.'
13055  103752  042101  051104  051505
13056  103760  026123  052600  044523
13057  103766  043516  052040  042510
13058  103774  046440  044501  052116
13059  104002  047105  047101  042503
13060  104010  051040  043505  051511
```

```
 13061  104016  042524  027122     000
 13062
 13063  104023                            EM140:
 13064  104023     115  044501  020116             .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 13065  104030  042515  047515  054522
 13066  104036  042040  052101  020101
 13067  104044  040520  044522  054524
 13068  104052  041440  042510  045503
 13069  104060  051105  020123  042524
 13070  104066  052123  043040  044501
 13071  104074  042514  027104
 13072  104100  052600  040516  046102             .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 13073  104106  020105  047524  043040
 13074  104114  051117  042503  040440
 13075  104122  050040  051101  052111
 13076  104130  020131  051105  047522
 13077  104136  026122  052440  044523
 13078  104144  043516     040
 13079  104147     124  042510  046440             .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 13080  104154  044501  052116  047105
 13081  104162  047101  042503  051040
 13082  104170  043505  051511  042524
 13083  104176  026122     200
 13084  104201     101  020124  044124             .ASCII  'AT THE MAIN MEMORY EVEN WORD, LOW BYTE, PARITY '
 13085  104206  020105  040515  047111
 13086  104214  046440  046505  051117
 13087  104222  020131  053105  047105
 13088  104230  053440  051117  026104
 13089  104236  046040  053517  041040
 13090  104244  052131  026105  050040
 13091  104252  051101  052111  020131
 13092  104260  044103  041505  042513             .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 13093  104266  026122  020200  042522
 13094  104274  042101  047111  020107
 13095  104302  020101  040504  040524
 13096  104310  050040  052101  042524
 13097  104316  047122  053440  044510
 13098  104324  044103     040
 13099  104327     123  047510  046125             .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 13100  104334  020104  040510  042526
 13101  104342  041440  052501  042523
 13102  104350  020104  047101  042440
 13103  104356  051122  051117  000056
 13104
 13105  104364                            EM141:
 13106  104364  040515  047111  046440             .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 13107  104372  046505  051117  020131
 13108  104400  040504  040524  050040
 13109  104406  051101  052111  020131
 13110  104414  044103  041505  042513
 13111  104422  051522  052040  051505
 13112  104430  020124  040506  046111
 13113  104436  042105     056
 13114  104441     200  047125  041101             .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 13115  104446  042514  052040  020117
 13116  104454  047506  041522  020105
```

```
13117  104462  020101  040520  044522
13118  104470  054524  042440  051122
13119  104476  051117  020054  051525
13120  104504  047111  020107
13121  104510  044124  020105  040515                 .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
13122  104516  047111  042524  040516
13123  104524  041516  020105  042522
13124  104532  044507  052123  051105
13125  104540  100054
13126  104542  052101  052040  042510                 .ASCII  'AT THE MAIN MEMORY ODD WORD, LOW BYTE, PARITY '
13127  104550  046440  044501  020116
13128  104556  042515  047515  054522
13129  104564  047440  042104  053440
13130  104572  051117  026104  046040
13131  104600  053517  041040  052131
13132  104606  026105  050040  051101
13133  104614  052111  020131
13134  104620  044103  041505  042513                 .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
13135  104626  026122  020200  042522
13136  104634  042101  047111  020107
13137  104642  020101  040504  040524
13138  104650  050040  052101  042524
13139  104656  047122  053440  044510
13140  104664  044103     040
13141  104667     123  047510  046125                 .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
13142  104674  020104  040510  042526
13143  104702  041440  052501  042523
13144  104710  020104  047101  042440
13145  104716  051122  051117  000056
13146
13147  104724                                EM142:
13148  104724  040515  047111  046440                 .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
13149  104732  046505  051117  020131
13150  104740  040504  040524  050040
13151  104746  051101  052111  020131
13152  104754  044103  041505  042513
13153  104762  051522  052040  051505
13154  104770  020124  040506  046111
13155  104776  042105     056
13156  105001     200  047125  041101                 .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
13157  105006  042514  052040  020117
13158  105014  047506  041522  020105
13159  105022  020101  040520  044522
13160  105030  054524  042440  051122
13161  105036  051117  020054  051525
13162  105044  047111  020107
13163  105050  044124  020105  040515                 .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
13164  105056  047111  042524  040516
13165  105064  041516  020105  042522
13166  105072  044507  052123  051105
13167  105100  100054
13'68  105102  052101  052040  042510                 .ASCII  'AT THE MAIN MEMORY EVEN WORD, HIGH BYTE, PARITY '
13169  105110  046440  044501  020116
13170  105116  042515  047515  054522
13171  105124  042440  042526  020116
13172  105132  047527  042122  020054
```

```
 13173  105140  044510  044107  041040
 13174  105146  052131  026105  050040
 13175  105154  051101  052111  020131
 13176  105162  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 13177  105170  026122  020200  042522
 13178  105176  042101  047111  020107
 13179  105204  020101  040504  040524
 13180  105212  050040  052101  042524
 13181  105220  047122  053440  044510
 13182  105226  044103     040
 13183  105231     123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN FRROR.'
 13184  105236  020104  040510  042526
 13185  105244  041440  052501  042523
 13186  105252  020104  047101  042440
 13187  105260  051122  051117  000056
 13188
 13189  105266                          EM143:
 13190  105266  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 13191  105274  046505  051117  020131
 13192  105302  040504  040524  050040
 13193  105310  051101  052111  020131
 13194  105316  044103  041505  042513
 13195  105324  051522  052040  051505
 13196  105332  020124  040506  046111
 13197  105340  042105     056
 13198  105343     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 13199  105350  042514  052040  020117
 13200  105356  047506  041522  020105
 13201  105364  020101  040520  044522
 13202  105372  054524  042440  051122
 13203  105400  051117  020054  051525
 13204  105406  047111  020107
 13205  105412  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 13206  105420  047111  042524  040516
 13207  105426  041516  020105  042522
 13208  105434  044507  052123  051105
 13209  105442  100054
 13210  105444  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY ODD WORD, HIGH BYTE, PARITY '
 13211  105452  046440  044501  020116
 13212  105460  042515  047515  054522
 13213  105466  047440  042104  053440
 13214  105474  051117  026104  044040
 13215  105502  043511  020110  054502
 13216  105510  042524  020054  040520
 13217  105516  044522  054524     040
 13218  105523     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 13219  105530  051105  100054  051040
 13220  105536  040505  044504  043516
 13221  105544  040440  042040  052101
 13222  105552  020101  040520  052124
 13223  105560  051105  020116  044127
 13224  105566  041511  020110
 13225  105572  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 13226  105600  044040  053101  020105
 13227  105606  040503  051525  042105
 13228  105614  040440  020116  051105
```

```
105622  047522  027122     000

105627                             EM144:
105627     103  041501  042510             .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
105634  042040  052101  020101
105642  042515  047515  054522
105650  050040  051101  052111
105656  020131  044103  041505
105664  042513  051522  052040
105672  051505  020124  040506
105700  046111  042105     056
105705     200  047125  041101             .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
105712  042514  052040  020117
105720  047506  041522  020105
105726  020101  040520  044522
105734  054524  042440  051122
105742  051117  020054  051525
105750  047111  020107
105754  044124  020105  040515             .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
105762  047111  042524  040516
105770  041516  020105  042522
105776  044507  052123  051105
106004  100054
106006  052101  052040  042510             .ASCII  'AT THE GROUP ZERO,LOW BYTE, DATA PARITY CHECKER,'
106014  043440  047522  050125
106022  055040  051105  026117
106030  047514  020127  054502
106036  042524  020054  040504
106044  040524  050040  051101
106052  052111  020131  044103
106060  041505  042513  026122
106066  051200  040505  044504             .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
106074  043516  040440  042040
106102  052101  020101  040520
106110  052124  051105  020116
106116  044127  041511  020110
106124  044123  052517  042114
106132  044040  053101  020105
106140  040503  051525  042105             .ASCIZ  'CAUSED AN ERROR.'
106146  040440  020116  051105
106154  047522  027122     000

106161                             EM145:
106161     103  041501  042510             .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
106166  042040  052101  020101
106174  042515  047515  054522
106202  050040  051101  052111
106210  020131  044103  041505
106216  042513  051522  052040
106224  051505  020124  040506
106232  046111  042105     056
106237     200  047125  041101             .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
106244  042514  052040  020117
106252  047506  041522  020105
106260  020101  040520  044522
106266  054524  042440  051122
```

```
106274  051117  020054  051525
106302  047111  020107
106306  044124  020105  040515               .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
106314  047111  042524  040516
106322  041516  020105  042522
106330  044507  052123  051105
106336  100054
106340  052101  052040  042510               .ASCII  'AT THE GROUP ONE,LOW BYTE, DATA PARITY CHECKER,'
106346  043440  047522  050125
106354  047440  042516  046054
106362  053517  041040  052131
106370  026105  042040  052101
106376  020101  040520  044522
106404  054524  041440  042510
106412  045503  051105     054
106417     200  042522  042101               .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
106424  047111  020107  020101
106432  040504  040524  050040
106440  052101  042524  047122
106446  053440  044510  044103
106454  051440  047510  046125
106462  020104  040510  042526
106470     040
106471     103  052501  042523               .ASCIZ  'CAUSED AN ERROR.'
106476  020104  047101  042440
106504  051122  051117  000056

106512                                EM146:  .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
106512  040503  044103  020105
106520  040504  040524  046440
106526  046505  051117  020131
106534  040520  044522  054524
106542  041440  042510  045503
106550  051105  020123  042524
106556  052123  043040  044501
106564  042514  027104
106570  052600  040516  046102               .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
106576  020105  047524  043040
106604  051117  042503  040440
106612  050040  051101  052111
106620  020131  051105  047522
106626  026122  052440  044523
106634  043516     040
106637     124  042510  046440               .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
106644  044501  052116  047105
106652  047101  042503  051040
106660  043505  051511  042524
106666  026122     200
106671     101  020124  044124               .ASCII  'AT THE GROUP ZERO,HIGH BYTE, DATA PARITY CHECKER,'
106676  020105  051107  052517
106704  020120  042532  047522
106712  044054  043511  020110
106720  054502  042524  020054
106726  040504  040524  050040
106734  051101  052111  020131
106742  044103  041505  042513
```

```
106750  026122
106752  051200  040505  044504                 .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
106760  043516  040440  042040
106766  052101  020101  040520
106774  052124  051105  020116
107002  044127  041511  020110
107010  044123  052517  042114
107016  044040  053101  020105
107024  040503  051525  042105                 .ASCIZ  'CAUSED AN ERROR.'
107032  040440  020116  051105
107040  047522  027122     000

107045                                 EM147:  .ASCII  'CACHE DATA MEMORY PARITY CHECKERS TEST FAILED.'
107045     103  041501  042510
107052  042040  052101  020101
107060  042515  047515  054522
107066  050040  051101  052111
107074  020131  044103  041505
107102  042513  051522  052040
107110  051505  020124  040506
107116  046111  042105     056
107123     200  047125  041101                 .ASCII  <CRLF>'UNABLE TO FORCE A PARITY ERROR, USING '
107130  042514  052040  020117
107136  047506  041522  020105
107144  020101  040520  044522
107152  054524  042440  051122
107160  051117  020054  051525
107166  047111  020107
107172  044124  020105  040515                 .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
107200  047111  042524  040516
107206  041516  020105  042522
107214  044507  052123  051105
107222  100054
107224  052101  052040  042510                 .ASCII  'AT THE GROUP ONE,HIGH BYTE, DATA PARITY CHECKER,'
107232  043440  047522  050125
107240  047440  042516  044054
107246  043511  020110  054502
107254  042524  020054  040504
107262  040524  050040  051101
107270  052111  020131  044103
107276  041505  042513  026122
107304  051200  040505  044504                 .ASCII  <CRLF>'READING A DATA PATTERN WHICH SHOULD HAVE '
107312  043516  040440  042040
107320  052101  020101  040520
107326  052124  051105  020116
107334  044127  041511  020110
107342  044123  052517  042114
107350  044040  053101  020105
107356  040503  051525  042105                 .ASCIZ  'CAUSED AN ERROR.'
107364  040440  020116  051105
107372  047522  027122     000

107377     200  047125  054105         EM150:  .ASCIZ  <CRLF>'UNEXPECTED CPU ERROR TRAPPED TO VECTOR ERRVEC (4) '
107404  042520  052103  042105
107412  041440  052520  042440
107420  051122  051117  052040
```

```
 13397  107426  040522  050120  042105
 13398  107434  052040  020117  042526
 13399  107442  052103  051117  042440
 13400  107450  051122  042526  020103
 13401  107456  032050  020451     000
 13402
 13403  107463     115  051501  020123  EM151:  .ASCIZ  'MASS BUS WRITE HIT DID NOT INVALIDATE THE CACHE.'
 13404  107470  052502  020123  051127
 13405  107476  052111  020105  044510
 13406  107504  020124  044504  020104
 13407  107512  047516  020124  047111
 13408  107520  040526  044514  040504
 13409  107526  042524  052040  042510
 13410  107534  041440  041501  042510
 13411  107542  000056
 13412
 13413          107463                  EM152=EM151
 13414          107463                  EM153=EM151
 13415
 13416  107544  042504  044526  042503  EM154:  .ASCIZ  'DEVICE ERROR IN THE RS04.'
 13417  107552  042440  051122  051117
 13418  107560  044440  020116  044124
 13419  107566  020105  051522  032060
 13420  107574  000056
 13421
 13422  107576  042504  044526  042503  EM155:  .ASCIZ  'DEVICE ERROR IN THE RP04.'
 13423  107604  042440  051122  051117
 13424  107612  044440  020116  044124
 13425  107620  020105  050122  032060
 13426  107626  000056
 13427
 13428  107630  042504  044526  042503  EM156:  .ASCIZ  'DEVICE ERROR IN THE MASS BUS TESTER.'
 13429  107636  042440  051122  051117
 13430  107644  044440  020116  044124
 13431  107652  020105  040515  051523
 13432  107660  041040  051525  052040
 13433  107666  051505  042524  027122
 13434  107674     000
 13435
 13436
 13437  107675     104  053105  041511  EM160:  .ASCIZ  'DEVICE ERROR IN THE RK05.'
 13438  107702  020105  051105  047522
 13439  107710  020122  047111  052040
 13440  107716  042510  051040  030113
 13441  107724  027065     000
 13442
 13443  107727     104  053105  041511  EM161:  .ASCIZ  'DEVICE ERROR IN THE UNIBUS EXERCISER.'
 13444  107734  020105  051105  047522
 13445  107742  020122  047111  052040
 13446  107750  042510  052440  044516
 13447  107756  052502  020123  054105
 13448  107764  051105  044503  042523
 13449  107772  027122     000
 13450  107775     125  041502  020102  EM162:  .ASCII  /UBCB PE ABORT DOESN'T GO LOW OR/<CRLF>
 13451  110002  042520  040440  047502
 13452  110010  052122  042040  042517
```

```
110016  047123  052047  043440
110024  020117  047514  020127
110032  051117     200
110035     111  020124  047504           .ASCII  /IT DOESN'T GET TO TMCC F33 OR E33 BAD/<CRLF>
110042  051505  023516  020124
110050  042507  020124  047524
110056  052040  041515  020103
110064  031505  020063  051117
110072  042440  031463  041040
110100  042101     200
110103     117  020122  041125           .ASCII  /OR UBCB PARITY ERR DOESN'T GET TO TMCB E53/<CRLF>
110110  041103  050040  051101
110116  052111  020131  051105
110124  020122  047504  051505
110132  023516  020124  042507
110140  020124  047524  052040
110146  041515  020102  032505
110154  100063
110156  051501  040440  046040           .ASCIZ  /AS A LOW OR E53(5) BAD/
110164  053517  047440  020122
110172  032505  024063  024465
110200  041040  042101     000
110205     125  041502  020102  EM163:   .ASCII  /UBCB PARITY ERR DOESN'T GO LOW OR IT DOES/<CRLF>
110212  040520  044522  054524
110220  042440  051122  042040
110226  042517  047123  052047
110234  043440  020117  047514
110242  020127  051117  044440
110250  020124  047504  051505
110256     200
110257     116  052117  043440           .ASCIZ  /NOT GET TO DAPE/
110264  052105  052040  020117
110272  040504  042520     000
110277     104  050101  020105  EM164:   .ASCIZ  /DAPE E11(4) BAD OR TV06 DOESN'T GET TO THE ALU/
110304  030505  024061  024464
110312  041040  042101  047440
110320  020122  053124  033060
110326  042040  042517  047123
110334  052047  043440  052105
110342  052040  020117  044124
110350  020105  046101  000125
110356  040504  042520  042440  EM165:   .ASCIZ  /DAPE E7(1) BAD/
110364  024067  024461  041040
110372  042101     000
110375     124  041515  020101  EM166:   .ASCIZ  /TMCA SEG+CON+PAR DOESN'T GO LOW ON CCBJ PARITY TRAP/
110402  042523  025507  047503
110410  025516  040520  020122
110416  047504  051505  023516
110424  020124  047507  046040
110432  053517  047440  020116
110440  041503  045102  050040
110446  051101  052111  020131
110454  051124  050101     000
110461     124  041515  020102  EM167:   .ASCII  /TMCB PART DOESN'T GO LOW OR DOES/<CRLF>
110466  040520  052122  042040
110474  042517  047123  052047
```

```
 13509  110502  043440  020117  047514
 13510  110510  020127  051117  042040
 13511  110516  042517  100123
 13512  110522  047516  020124  042507           .ASCIZ  /NOT GET TO UBCB OR UBCB E18(1) BAD/
 13513  110530  020124  047524  052440
 13514  110536  041502  020102  051117
 13515  110544  052440  041502  020102
 13516  110552  030505  024070  024461
 13517  110560  041040  042101     000
 13518  110565     124  041515  020101  EM170:  .ASCIZ  /TMCA E67(8) DOESN'T GO LOW ON MGMT/
 13519  110572  033105  024067  024470
 13520  110600  042040  042517  047123
 13521  110606  052047  043440  020117
 13522  110614  047514  020127  047117
 13523  110622  046440  046507  000124
 13524  110630  046524  040503  042440  EM171:  .ASCIZ  /TMCA E67(12) DOESN'T GO LOW ON MGMT/
 13525  110636  033466  030450  024462
 13526  110644  042040  042517  047123
 13527  110652  052047  043440  020117
 13528  110660  047514  020127  047117
 13529  110666  046440  046507  000124
 13530  110674  046524  040503  042440  EM172:  .ASCII  /TMCA E68(6) DOESN'T GO LOW ON PAR TRP/<CRLF>
 13531  110702  034066  033050  020051
 13532  110710  047504  051505  023516
 13533  110716  020124  047507  046040
 13534  110724  053517  047440  020116
 13535  110732  040520  020122  051124
 13536  110740  100120
 13537  110742  051117  042440  032464           .ASCIZ  /OR E45(4) BAD/
 13538  110750  032050  020051  040502
 13539  110756  000104
 13540  110760  046524  040503  042440  EM173:  .ASCIZ  /TMCA E68(8) DOESN'T GO LOW ON PAR TRP/
 13541  110766  034066  034050  020051
 13542  110774  047504  051505  023516
 13543  111002  020124  047507  046040
 13544  111010  053517  047440  020116
 13545  111016  040520  020122  051124
 13546  111024  000120
 13547  111026  046524  041503  050040  EM435:  .ASCII  /TMCC PRIORITY CLEAR DIDN'T GO LOW OR DIDN'T/<CRLF>
 13548  111034  044522  051117  052111
 13549  111042  020131  046103  040505
 13550  111050  020122  044504  047104
 13551  111056  052047  043440  020117
 13552  111064  047514  020127  051117
 13553  111072  042040  042111  023516
 13554  111100  100124
 13555  111102  042507  020124  044124           .ASCIZ  /GET THRU TMCA E43(2) ON ABORT CLEAR/
 13556  111110  052522  052040  041515
 13557  111116  020101  032105  024063
 13558  111124  024462  047440  020116
 13559  111132  041101  051117  020124
 13560  111140  046103  040505  000122
 13561  111146  052502  020123  041120  EM174:  .ASCIZ  /BUS PB DIDN'T GET TO UBCB PE ABORT/
 13562  111154  042040  042111  023516
 13563  111162  020124  042507  020124
 13564  111170  047524  052440  041502
```

```
111176  020102  042520  040440
111204  047502  052122     000
111211     125  041502  020102  EM175:  .ASCIZ  /UBCB PARITY ERR DIDN'T GO LOW ON BUS PB/
111216  040520  044522  054524
111224  042440  051122  042040
111232  042111  023516  020124
111240  047507  046040  053517
111246  047440  020116  052502
111254  020123  041120     000

                                ;THESE ARE DATA HEADERS:

111261     040  052040  051505  DH1:    .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'PHYSICAL ADDR.'<TAB>'CALL AT PC.'
111266  027124  020011  051107
111274  052517  027120  050011
111302  054510  044523  040503
111310  020114  042101  051104
111316  004456  040503  046114
111324  040440  020124  041520
111332  000056

111334  020040  042524  052123  DH2:    .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR ADDR REG.'<TAB>'ERROR REG.'<TAB>
111342  004456  043440  047522
111350  050125  004456  051105
111356  047522  020122  042101
111364  051104  051040  043505
111372  004456  051105  047522
111400  020122  042522  027107
111406     011
111407     122  043105  040440          .ASCIZ  'REF ADDR.'<TAB>'TRAP AT PC.'
111414  042104  027122  052011
111422  040522  020120  052101
111430  050040  027103     000

        111334                  DH3=DH2

        111334                  DH4=DH2

111435     040  052040  051505  DH5:    .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'READ.'
111442  027124  041411  046101
111450  020114  052101  050040
111456  027103  051011  040505
111464  027104     000

        111435                  DH6=DH5

        111435                  DH7-DH5

        111435                  DH10-DH5

111467     040  052040  051505  DH11:   .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'TRAP AT PC.'<TAB>'ERROR ADDR REG.'
111474  027124  020011  051107
111502  052517  027120  052011
111510  040522  020120  052101
111516  050040  027103  042411
111524  051122  051117  040440
```

```
13621  111532  042104  020122  042522
13622  111540  027107     000
13623
13624  111543     040  052040  051505  DH12:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'CALL AT PC.'<TAB>'TEST ADDR.'<TAB>
13625  111550  027124  020011  051107
13626  111556  052517  027120  041411
13627  111564  046101  020114  052101
13628  111572  050040  027103  052011
13629  111600  051505  020124  042101
13630  111606  051104  004456
13631  111612  040504  040524  053440          .ASCIZ  'DATA WR. DATA READ.'
13632  111620  027122  042040  052101
13633  111626  020101  042522  042101
13634  111634  000056
13635
13636  111636  020040  042524  052123  DH13:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'*DATA.'<TAB>'+DATA.'<TAB>
13637  111644  004456  043440  047522
13638  111652  050125  004456  042052
13639  111660  052101  027101  025411
13640  111666  040504  040524  004456
13641  111674  051105  047522  020122          .ASCIZ  'ERROR COUNT.'
13642  111702  047503  047125  027124
13643  111710     000
13644
13645  111711     040  052040  051505  DH14:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADDR REG.'
13646  111716  027124  041411  046101
13647  111724  020114  052101  050040
13648  111732  027103  042411  051122
13649  111740  051117  040440  042104
13650  111746  020122  042522  027107
13651  111754  052011  040522  020120          .ASCII  <TAB>'TRAP AT PC.'<TAB>
13652  111762  052101  050040  027103
13653  111770     011
13654  111771     105  051122  051117          .ASCIZ  'ERROR REG.'
13655  111776  051040  043505  000056
13656
13657  112004  020040  042524  052123  DH15:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
13658  112012  004456  040503  046114
13659  112020  040440  020124  041520
13660  112026  000056
13661
13662  112030  020040  042524  052123  DH16:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'WROTE.'<TAB>'READ.'<TAB>
13663  112036  004456  043440  047522
13664  112044  050125  004456  051127
13565  112052  052117  027105  051011
13666  112060  040505  027104     011
13667  112065     101  042104  020122          .ASCIZ  'ADDR TESTED.'<TAB>'CALL AT PC.'
13668  112072  042524  052123  042105
13669  112100  004456  040503  046114
13670  112106  040440  020124  041520
13671  112114  000056
13672
13673  112116  020040  042524  052123  DH17:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR AT PC.'<TAB>'READ.'<TAB>
13674  112124  004456  043440  047522
13675  112132  050125  004456  051105
13676  112140  047522  020122  052101
```

```
112146  050040  027103  051011
112154  040505  027104     011
112161     111  027116  040411          .ASCIZ  'IN.'<TAB>'ADDRESS.'
112166  042104  042522  051523
112174  000056

        112116                  DH20=DH17

112176  020040  042524  052123  DH21:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'READ.'<TAB>' GROUP.'<TAB>'ADDRESS.'
112204  004456  040503  046114
112212  040440  020124  041520
112220  004456  042522  042101
112226  004456  043440  047522
112234  050125  004456  042101
112242  051104  051505  027123
112250     000

112251     040  052040  051505  DH22:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'EXPECTED ERROR AT.'
112256  027124  041411  046101
112264  020114  052101  050040
112272  027103  042411  050130
112300  041505  042524  020104
112306  051105  047522  020122
112314  052101  000056

112320  020040  042524  052123  DH23:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'EXPECTED ADRS.'<TAB>
112326  004456  040503  046114
112334  040440  020124  041520
112342  004456  054105  042520
112350  052103  042105  040440
112356  051104  027123     011
112363     107  052117  040440          .ASCIZ  'GOT ADRS.'<TAB>'ERROR REG.'
112370  051104  027123  042411
112376  051122  051117  051040
112404  043505  000056

        112251                  DH24=DH22

        112320                  DH25=DH23

112410  020040  042524  052123  DH26:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
112416  004456  040503  046114
112424  040440  020124  041520
112432  004456  043440  047522
112440  050125  004456  042101
112446  051104  051505  027123
112454     000

112455     040  052040  051505  DH27:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ESTABLISHED HIT.'
112462  027124  041411  046101
112470  020114  052101  050040
112476  027103  020011  051107
112504  052517  027120  042411
112512  052123  041101  044514
112520  044123  042105  044040
112526  052111     056
```

```
13733 112531    040 052502 020124          .ASCIZ  ' BUT GOT HIT.'
13734 112536 047507 020124 044510
13735 112544 027124    000
13736
13737        112251                     DH30=DH22
13738
13739        112320                     DH31=DH23
13740
13741        112251                     DH32=DH22
13742
13743        112320                     DH33=DH23
13744
13745 112547    040 052040 051505  DH34:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'READ.'<TAB>'IN ADDRESS.'<TAB>
13746 112554 027124 050011 020103
13747 112562 043117 041440 046101
13748 112570 027114 051011 040505
13749 112576 027104 044411 020116
13750 112604 042101 051104 051505
13751 112612 027123    011
13752 112615    105 050130 041505          .ASCIZ  'EXPECTED.'
13753 112622 042524 027104    000
13754
13755        112547                     DH35=DH34
13756
13757 112627    040 052040 051505  DH36:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
13758 112634 027124 041411 046101
13759 112642 020114 052101 050040
13760 112650 027103 020011 051107
13761 112656 052517 027120 040411
13762 112664 042104 042522 051523
13763 112672 000056
13764
13765 112674 020040 042524 052123  DH37:   .ASCII  '  TEST.'<TAB>' GROUP.'<TAB>'ERROR COUNT.'<TAB>
13766 112702 004456 043440 047522
13767 112710 050125 004456 051105
13768 112716 047522 020122 047503
13769 112724 047125 027124    011
13770 112731    052 041040 042101          .ASCIZ  '* BAD ADRS.'<TAB>'+ BAD ADRS.'
13771 112736 040440 051104 027123
13772 112744 025411 041040 042101
13773 112752 040440 051104 027123
13774 112760    000
13775
13776
13777 112761    040 052040 051505  DH41:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
13778 112766 027124 041411 046101
13779 112774 020114 052101 050040
13780 113002 027103 020011 051107
13781 113010 052517 027120 040411
13782 113016 042104 042522 051523
13783 113024 000056
13784
13785        112761                     DH42-DH41
13786
13787 113026 020040 042524 052123  DH43:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>' GROUP.'
13788 113034 004456 040503 046114
```

```
13789  113042  040440  020124  041520
13790  113050  004456  051124  050101
13791  113056  040440  020124  041520
13792  113064  004456  043440  047522
13793  113072  050125     056
13794
13795          113026                   DH40=DH43
13796
13797  113075     040  052040  051505   DH44:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
13798  113102  027124  041411  046101
13799  113110  020114  052101  050040
13800  113116  027103  052011  040522
13801  113124  020120  052101  050040
13802  113132  027103     011
13803  113135     105  051122  051117           .ASCIZ  'ERROR ADRS REG.'<TAB>'ERROR REG.'
13804  113142  040440  051104  020123
13805  113150  042522  027107  042411
13806  113156  051122  051117  051040
13807  113164  043505  000056
13808
13809          113075                   DH45=DH44
13810
13811  113170  020040  042524  052123   DH46:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
13812  113176  004456  040503  046114
13813  113204  040440  020124  041520
13814  113212  000056
13815
13816          113170                   DH47=DH46
13817
13818          113075                   DH50=DH44
13819
13820          113075                   DH51=DH44
13821
13822          113170                   DH52=DH46
13823
13824          113170                   DH53=DH46
13625
13826  113214  020040  042524  052123   DH54:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR COUNT.'
13827  113222  004456  040503  046114
13828  113230  040440  020124  041520
13829  113236  004456  051105  047522
13830  113244  020122  047503  047125
13831  113252  027124     000
13832  113255     040  050040  020103   DH55:   .ASCIZ  /  PC    CCR/
13833  113262  020040  041440  051103
13834  113270     000
13835  113271     040  050040  020103   DH66:   .ASCIZ  /  PC    CCR   GROUP   TST-DATA-ADRS/
13836  113276  020040  041440  051103
13837  113304  020040  043440  047522
13838  113312  050125  020040  052040
13839  113320  052123  042055  052101
13840  113326  026501  042101  051522
13841  113334     000
13842  113335     040  050040  020103   DH71:   .ASCIZ  /  PC   EXPCTD  RECVD  LOC/
13843  113342  020040  054105  041520
13844  113350  042124  020040  042522
```

```
113356 053103 020104 046040
113364 041517    000
113367    040 050040 020103   DH107:  .ASCIZ  /  PC     CCR    MSER    GROUP    EXPCTD-B4,5/
113374 020040 041440 051103
113402 020040 020040 051515
113410 051105 020040 043440
113416 047522 050125 020040
113424 042440 050130 052103
113432 026504 032102 032454
113440    000
113441    040 050040 020103   DH111:  .ASCIZ  /  PC     MSER/
113446 020040 046440 042523
113454 000122
113456 020040 041520 020040   DH115:  .ASCIZ  /  PC     HITMIS/
113464 020040 044510 046524
113472 051511    000

113475    040 050040 020103   DH123:  .ASCIZ  ?  PC     KIPDR  (KIPDR)?
113502 020040 045440 050111
113510 051104 020040 045450
113516 050111 051104 000051
113524 020040 041520 020040   DH125:  .ASCIZ  /  PC     CCR  PAR-ADR   (PAR)  (PDR)  TST-DATA-ADRS(VA)/
113532 020040 041503 020122
113540 050040 051101 040455
113546 051104 020040 024040
113554 040520 024522 020040
113562 050050 051104 020051
113570 052040 052123 042055
113576 052101 026501 042101
113604 051522 053050 024501
113612    000
113613    040 050040 020103   DH130:  .ASCIZ  ?  PC     SIPDR  (SIPDR)?
113620 020040 051440 050111
113626 051104 020040 051450
113634 050111 051104 000051
113642 020040 041520 020040   DH132:  .ASCIZ  ?  PC     UIPDR  (UIPDR)?
113650 020040 044525 042120
113656 020122 024040 044525
113664 042120 024522    000


113671    040 052040 051505   DH136:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>'ADDRESS.'
113676 027124 041411 046101
113704 020114 052101 050040
113712 027103 020011 051107
113720 052517 027120 040411
113726 042104 042522 051523
113734 000056

       113671                 DH137=DH136

113736 020040 042524 052123   DH140:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'DATA.'<TAB>'ADDRESS.'
113744 004456 040503 046114
113752 040440 020124 041520
113760 004456 040504 040524
113766 004456 042101 051104
```

```
113774  051505  027123     000

        113736                      DH141=DH140

        113736                      DH142=DH140

        113736                      DH143=DH140

        113736                      DH144=DH140

        113736                      DH145=DH140

        113736                      DH146=DH140

        113736                      DH147=DH140

114001     040  052040  051505   DH150:  .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'CPU ERROR REGISTER.'
114006  027124  052011  040522
114014  020120  052101  050040
114022  027103  041411  046101
114030  020114  052101  050040
114036  027103  041411  052520
114044  042440  051122  051117
114052  051040  043505  051511
114060  042524  027122     000

114065     125  044523  043516   DH151:  .ASCII  'USING THE RS04.'
114072  052040  042510  051040
114100  030123  027064
114104  020040  042524  052123           .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
114112  004456  051107  052517
114120  027120  040411  042104
114126  042522  051523  000056

114134  051525  047111  020107   DH152:  .ASCII  'USING THE RP04.'
114142  044124  020105  050122
114150  032060     056
114153     040  052040  051505           .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
114160  027124  043411  047522
114166  050125  004456  042101
114174  051104  051505  027123
114202     000

114203     125  044523  043516   DH153:  .ASCII  'USING THE MASS BUS TESTER.'
114210  052040  042510  046440
114216  051501  020123  052502
114224  020123  042524  052123
114232  051105     056
114235     040  052040  051505           .ASCIZ  '  TEST.'<TAB>'GROUP.'<TAB>'ADDRESS.'
114242  027124  043411  047522
114250  050125  004456  042101
114256  051104  051505  027123
114264     000

114265     040  052040  051505   DH154:  .ASCIZ  '  TEST.'<TAB>'RS4CS2.'<TAB>'RS4DS.'<TAB>'RS4ER.'
114272  027124  051011  032123
```

```
114300  051503  027062  051011
114306  032123  051504  004456
114314  051522  042464  027122
114322     000

114323     040  052040  051505  DH155:  .ASCIZ  '  TEST.'<TAB>'RP4CS2.'<TAB>'RP4DS.'<TAB>'RP4ER.'
114330  027124  051011  032120
114336  051503  027062  051011
114344  032120  051504  004456
114352  050122  042464  027122
114360     000

114361     040  052040  051505  DH156:  .ASCIZ  '  TEST.'<TAB>'RH4CS2.'<TAB>'RH4ST.'<TAB>'RH4ER.'
114366  027124  051011  032110
114374  051503  027062  051011
114402  032110  052123  004456
114410  044122  042464  027122
114416     000


114417     040  052040  051505  DH160:  .ASCIZ  '  TEST.'<TAB>'RK5ER.'<TAB>'RK5DS.'
114424  027124  051011  032513
114432  051105  004456  045522
114440  042065  027123     000

114445     040  052040  051505  DH161:  .ASCIZ  '  TEST.'<TAB>'UBECR1.'<TAB>'UBECR2.'
114452  027124  052411  042502
114460  051103  027061  052411
114466  042502  051103  027062
114474     000
114475     105  051122  051117  DH162:  .ASCIZ  /ERRORPC TEST NUMBER/
114502  041520  052040  051505
114510  020124  052516  041115
114516  051105     000

                                :THESE ARE DATA FORMAT DESIGNATORS FOR THE DATA TABLE:

114521     004     004     003  DF1:    .BYTE   4,4,3,3
114524     003

114525     004     004     007  DF2:    .BYTE   4,4,7,0,3,3
114530     000     003     003

        114525                  DF3-DF2

        114525                  DF4=DF2

114533     004     003     000  DF5:    .BYTE   4,3,0,5,0,0,0,0
114536     005     000     000
114541     000     000

        114533                  DF6=DF5

        114533                  DF7-DF5

        114533                  DF10=DF5
```

```

 114543     004      004      003  DF11:    .BYTE   4,4,3,7,5,0,5,3,0
 114546     007      005      000
 114551     005      003      000

 114554     004      004      003  DF12:    .BYTE   4,4,3,3,0,0
 114557     003      000      000

 114562     004      004      000  DF13:    .BYTE   4,4,0,0,4
 114565     000      004

 114567     004      003      007  DF14:    .BYTE   4,3,7,3,0
 114572     003      000

 114574     004      003           DF15:    .BYTE   4,3

 114576     004      004      000  DF16:    .BYTF   4,4,0,0,3,3
 114601     000      003      003

 114604     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
 114607     000      005      003
 114612     005      005      005
 114615     003      005      003
 114620     005      003      005
 114623     003      005      000
 114626     005      000      005
 114631     000      005      000

         114604                    DF20=DF17

 114634     004      003      000  DF21:    .BYTE   4,3,0,4,3,5
 114637     004      003      005           .BYTE   5,3,5,3,5,3,5,3,5
 114642     005      003      005
 114645     003      005      003
 114650     005      003      005
 114653     000      005      000           .BYTE   0,5,0,5,0,5,0
 114656     005      000      005
 114661     000

 114662     004      003      002  DF22:    .BYTE   4,3,2

 114665     004      003      002  DF23:    .BYTE   4,3,2,2,0
 114670     002      000

         114662                    DF24=DF22

         114665                    DF25=DF23

 114672     004      003      004  DF26:    .BYTE   4,3,4,2
 114675     002

 114676     004      003      004  DF27:    .BYTE   4,3,4,2,2
 114701     002      002

         114662                    DF30-DF22

```

```
14069              114665                      DF31=DF23
14070
14071              114662                      DF32=DF22
14072
14073              114665                      DF33=DF23
14074
14075   114703       004       003       000   DF34:    .BYTE   4,3,0,2,0
14076   114706       002       000
14077
14078              114703                      DF35=DF34
14079
14080   114710       004       003       004   DF36:    .BYTE   4,3,4,2
14081   114713       002
14082
14083   114714       004       004       007   DF37:    .BYTE   4,4,7,2,2,0
14084   114717       002       002       000
14085
14086
14087   114722       004       003       004   DF41:    .BYTE   4,3,4,2
14088   114725       002
14089
14090              114722                      DF42=DF41
14091
14092   114726       004       003       003   DF43:    .BYTE   4,3,3,4,5,2,7,0
14093   114731       004       005       002
14094   114734       007       000
14095
14096              114726                      DF40=DF43
14097
14098   114736       004       003       002   DF44:    .BYTE   4,3,2,7,0,5,2,5,0,5,2,5,0,5,2
14099   114741       007       000       005
14100   114744       002       005       000
14101   114747       005       002       005
14102   114752       000       005       002
14103
14104              114736                      DF45=DF44
14105
14106   114755       004       003       005   DF46:    .BYTE   4,3,5,2,5,0,5,2,5,0,5,2
14107   114760       002       005       000
14108   114763       005       002       005
14109   114766       000       005       002
14110
14111              114755                      DF47=DF46
14112
14113              114736                      DF50=DF44
14114
14115              114736                      DF51=DF44
14116
14117              114755                      DF52=DF46
14118
14119              114755                      DF53=DF46
14120
14121   114771       004       003       004   DF54:    .BYTE   4,3,4
14122   114774       000       000             DF55:    .BYTE   0,0
14123              114774                      DF56=DF55
14124              114774                      DF57=DF55
```

```
14125          114774                     DF60=DF55
14126          114774                     DF61=DF55
14127          114774                     DF62=DF55
14128          114774                     DF63=DF55
14129          114774                     DF64=DF55
14130          114774                     DF65=DF55
14131  114776     000      000      000   DF66:   .BYTE   0,0,0,0
14132  115001     000
14133          114776                     DF67=DF66
14134          114776                     DF70=DF66
14135          114776                     DF71=DF66
14136          114776                     DF72=DF66
14137          114776                     DF73=DF66
14138          114776                     DF74=DF66
14139          114776                     DF75=DF66
14140          114776                     DF76=DF66
14141          114776                     DF77=DF66
14142
14143  115002     000      000      000   DF100:  .BYTE   0,0,0,0,0,0
14144  115005     000      000      000
14145          114774                     DF103=DF55
14146          114774                     DF104=DF55
14147          114774                     DF105=DF55
14148          114774                     DF106=DF55
14149  115010     000      000      000   DF107:  .BYTE   0,0,0,0,0
14150  115013     000      000
14151
14152          114774                     DF110=DF55
14153          114774                     DF111=DF55
14154          114774                     DF112=DF55
14155          114776                     DF113=DF66
14156          114774                     DF115=DF55
14157  115015     000      000      000   DF123:  .BYTE   0,0,0
14158  115020     004      003      004   DF136:  .BYTE   4,3,4,2
14159  115023     002
14160          115020                     DF137=DF136
14161
14162  115024     004      003      000   DF140:  .BYTE   4,3,0,2
14163  115027     002
14164
14165          115024                     DF141=DF140
14166
14167          115024                     DF142=DF140
14168
14169          115024                     DF143=DF140
14170
14171          115024                     DF144=DF140
14172
14173          115024                     DF145=DF140
14174
14175          115024                     DF146=DF140
14176
14177          115024                     DF147=DF140
14178
14179  115030     004      003      003   DF150:  .BYTE   4,3,3,0
14180  115033     000
```

```

115034     004     004     007  DF151:  .BYTE   4,4,7

        115034                  DF152=DF151
        115034                  DF153=DF151

115037     004     000     000  DF154:  .BYTE   4,0,0,0
115042     000

        115037                  DF155=DF154
        115037                  DF156=DF154
        115037                  DF157=DF154
        115037                  DF160=DF154
        115037                  DF161=DF154


        115044                          .EVEN

                                ;THESE ARE DATA TABLES:

115044  001632  001634  001636  DT1:    .WORD   $TMP0,$TMP1,$TMP2,$ERRPC,0
115052  001516  000000

115056  001632  001646  001636  DT2:    .WORD   $TMP0,$TMP6,$TMP2,$TMP1,$TMP5,$TMP4,0
115064  001634  001644  001642
115072  000000

        115056                  DT3=DT2

        115056                  DT4=DT2

115074  001632  001516  001636  DT5:    .WORD   $TMP0,$ERRPC,$TMP2,MTA5,JJPAT1,JJPAT2,JJPAT3,JJPAT4,0
115102  066502  026276  026300
115110  026302  026304  000000

        115074                  DT6=DT5

        115074                  DT7=DT5

        115074                  DT10=DT5

115116  001632  001634  001636  DT11:   .WORD   $TMP0,$TMP1,$TMP2,$TMP4,MTA11,$TMP3,$TAB,$TMP7,$TMP6,0
115124  001642  066564  001640
115132  066500  001650  001646
115140  000000

115142  001632  001634  001516  DT12:   .WORD   $TMP0,$TMP1,$ERRPC,$TMP3,$TMP4,$TMP5,0
115150  001640  001642  001644
115156  000000

115160  001632  001634  001636  DT13:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,0
115166  001640  001642  000000

115174  001632  001516  001634  DT14:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,0
115202  001640  001642  000000

```

```
115210  001632  001634  000000  DT15:   .WORD   $TMP0,$TMP1,0

115216  001632  001634  001636  DT16:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,$ERRPC,0
115224  001640  001642  001516
115232  000000

115234  001632  001634  001636  DT17:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,MTC17,$TMP4,$CRLF,MTB17
115242  001640  066656  001642
115250  001713  066636
115254  066631  001644  066631          .WORD   MTA17,$TMP5,MTA17,$TMP6,MTA17,$TMP7,MTA17,$TMP10
115262  001646  066631  001650
115270  066631  001652
115274  001713  035542  066500          .WORD   $CRLF,MMPAT1,$TAB,MMPAT2,$TAB,MMPAT3,$TAB,MMPAT4,0
115302  035544  066500  035546
115310  066500  035550  000000

115316  001632  001634  001636  DT20:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,MTA20,$TMP4,$CRLF,MTB17
115324  001640  066665  001642
115332  001713  066636
115336  001644  066631  001646          .WORD   $TMP5,MTA17,$TMP6,MTA17,$TMP7,MTA17,$TMP10,MTA17
115344  066631  001650  066631
115352  001652  066631
115356  001713  035542  066500          .WORD   $CRLF,MMPAT1,$TAB,MMPAT3,$TAB,MMPAT3,$TAB,MMPAT4,0
115364  035546  066500  035546
115372  066500  035550  000000

115400  001632  001634  001636  DT21:   .WORD   $TMP0,$TMP1,$TMP2,$TMP3,$TMP4,MTA21
115406  001640  001642  066674          .WORD   MTB21,$TMP5,MTB21,$TMP6,MTB21,$TMP7,MTB21,$TMP10,$CRLF
115414  066631  001644  066631
115422  001646  066631  001650
115430  066631  001652  001713
115436  033676  066500  033700          .WORD   KKPAT1,$TAB,KKPAT2,$TAB,KKPAT3,$TAB,KKPAT4,0
115444  066500  033702  066500
115452  033704  000000

115456  001632  001516  007632  DT22:   .WORD   $TMP0,$ERRPC,XADR2,0
115464  000000

115466  001632  001516  007632  DT23:   .WORD   $TMP0,$ERRPC,XADR2,$TMP3,$TMP1,0
115474  001640  001634  000000

115502  001632  001516  010532  DT24:   .WORD   $TMP0,$ERRPC,XXADR2,0
115510  000000

115512  001632  001516  010532  DT25:   .WORD   $TMP0,$ERRPC,XXADR2,$TMP3,$TMP1,0
115520  001640  001634  000000

115526  001632  001516  001634  DT26:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP2,0
115534  001636  000000

115540  001632  001516  001634  DT27:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP2,$TMP4,0
115546  001636  001642  000000

115554  001632  001516  011450  DT30:   .WORD   $TMP0,$ERRPC,RRADR2,0
115562  000000

```

```
115564  001632  001516  011450  DT31:   .WORD   $TMP0,$ERRPC,RRADR2,$TMP3,$TMP1,0
115572  001640  001634  000000

115600  001632  001516  012332  DT32:   .WORD   $TMP0,$ERRPC,SSADR2,0
115606  000000

115610  001632  001516  012332  DT33:   .WORD   $TMP0,$ERRPC,SSADR2,$TMP3,$TMP1,0
115616  001640  001634  000000

115624  001632  001516  001636  DT34:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,$TMP5,0
115632  001640  001644  000000

        115624                  DT35=DT34

115640  001632  001516  001636  DT36:   .WORD   $TMP0,$ERRPC,$TMP2,BBADR1,0
115646  015136  000000

115652  001632  001634  015156  DT37:   .WORD   $TMP0,$TMP1,BBCNT1,BBADR2,BBADR3,0
115660  015142  015146  000000


115666  001632  001516  001636  DT41:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
115674  001640  000000

        115666                  DT42=DT41

115700  001632  001516  001636  DT43:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA43,$TMP5,$TMP7,$TMP4,0
115706  001640  066761  001644
115714  001650  001642  000000

        115700                  DT40=DT43

115722  001632  001516  001666  DT44:   .WORD   $TMP0,$ERRPC,$TMP16,$TMP3,$TMP5,MTA45,$TMP12,MTB45
115730  001640  001644  067034
115736  001656  067062
115742  001652  067077  001646          .WORD   $TMP10,MTC45,$TMP6,MTB45,$TMP11,MTC45,$TMP14,0
115750  067062  001654  067077
115756  001662  000000

        115722                  DT45=DT44

115762  001632  001656  067034  DT46:   .WORD   $TMP0,$TMP12,MTA45,$TMP10,MTB45,$TMP6,MTC45
115770  001652  067062  001646
115776  067077
116000  001636  067062  001650          .WORD   $TMP2,MTB45,$TMP7,MTC45,$TMP4,0
116006  067077  001642  000000

        115762                  DT47=DT46

116014  001632  001516  001666  DT50:   .WORD   $TMP0,$ERRPC,$TMP16,$TMP3,$TMP5,MTA50,$TMP12,MTB45
116022  001640  001644  067112
116030  001656  067062
116034  001652  067077  001646          .WORD   $TMP10,MTC45,$TMP6,MTB45,$TMP11,MTC45,$TMP14,0
116042  067062  001654  067077
116050  001662  000000

```

```
                116014                      DT51=DT50

        116054  001632  001656  067112  DT52:   .WORD   $TMP0,$TMP12,MTA50,$TMP10,MTB45,$TMP6,MTC45
        116062  001652  067062  001646
        116070  067077
        116072  001636  067062  001650          .WORD   $TMP2,MTB45,$TMP7,MTC45,$TMP4,0
        116100  067077  001642  000000

                116054                      DT53=DT52

        116106  001632  001516  001636  DT54:   .WORD   $TMP0,$ERRPC,$TMP2,0
        116114  000000
        116116  001516  001562  000000  DT55:   .WORD   $ERRPC,$REG0,0
        116124  001516  001562  001564  DT66:   .WORD   $ERRPC,$REG0,$REG1,$REG2,0
        116132  001566  000000
        116136  001516  001562  001564  DT100:  .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,0
        116144  001566  001570  001572
        116152  000000
        116154  001516  001562  001564  DT107:  .WORD   $ERRPC, $REG0,  $REG1, $REG2, $REG3,0
        116162  001566  001570  000000

        116170  001516  001562  001564  DT123:  .WORD   $ERRPC,$REG0,$REG1,0
        116176  000000
        116200  001516  001562  001564  DT125:  .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,0
        116206  001566  001570  001572
        116214  000000

        116216  001632  001516  001636  DT136:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
        116224  001640  000000

                116216                      DT137=DT136

        116230  001632  001516  001636  DT140:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
        116236  001640  000000

                116230                      DT141=DT140

                116230                      DT142=DT140

                116230                      DT143=DT140

                116230                      DT144=DT140

                116230                      DT145=DT140

                116230                      DT146=DT140

                116230                      DT147=DT140

        116242  001632  001634  001636  DT150:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
        116250  001640  000000

        116254  001632  001634  001636  DT151:  .WORD   $TMP0,$TMP1,$TMP2,0
        116262  000000

                116254                      DT152-DT151
```

```
14405           116254                          DT153=DT151
14406   116264  001632  001634  001636  DT154:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
14407   116272  001640  000000
14408           116264                          DT155=DT154
14409           116264                          DT156=DT154
14410           116254                          DT157=DT151
14411           116254                          DT160=DT151
14412           116254                          DT161=DT151
14413   116276  001516  001502  000000  DT162:  .WORD   $ERRPC,$TSTNM,0
14414
14415           116304                                  TLOC=.
14416           116304                                  TLOC=-4&TLOC
14417           116310                                  TLOC=TLOC+4
14418           116310                                  .=TLOC
14419   116310  001000                          TSTDAT: .BLKW   512.
14420
14421
14422
14423   120310  000000  000000  000000  BOTTOM: .WORD   0,0,0
14424           126310                                  BOTPRG=BOTTOM+6000
14425           000001                                  .END
```

```
AA     = 000015          3414#
AAADR1   016360          3455*   3456*   3465    3466    3577*   3578*   3599#
AAADR2   016370          3465*   3466*   3467*   3468*   3469*   3479    3481    3502    3503    3555    3556    3573    3574
                         3603#   3624    3626
AADONE   016536          3582    3653#
AAERGS   016356          3452*   3542    3567    3586*   3598#
AAERR1   016404          3538    3610#
AAERR2   016474          3632    3641#
AAEXER   016374          3453*   3583*   3605#   3614
AAFLG1   016350          3451*   3530    3554    3572    3581    3585*   3589#
AAFLG2   016352          3496*   3565*   3592#   3645    3648
AAGS     016354          3450*   3459    3584*   3596#
AAHIAD   015572          3439#
AALOAD   015570          3438#   3440*   3479    3481
AAOFST   016364          3444*   3445*   3467    3469    3601#
AATMP1   016376          3606#
AATMP2   016400          3442    3607#
AA1      015664          3455#   3587
AA16     016244          3571#
AA2      015704          3462#   3579
AA3      015776          3487    3492    3496#
AA4      016146          3527    3536#
AA5      016174          3553#
AA6      016220          3559#   3650
AA7      016270          3534    3577#   3647
AA8      016306          3490    3581#
ABASE  = 000000           598
ABORTT   055544         10024   10255#
ACDW1  = 000000           598
ACDW2  = 000000           598
ACPUOP=  000000           598     613
ADDW0  = 000000           598
ADDW1  = 000000           598
ADDW10=  000000           598
ADDW11=  000000           598
ADDW12=  000000           598
ADDW13=  000000           598
ADDW14=  000000           598
ADDW15=  000000           598
ADDW2  = 000000           598
ADDW3  - 000000           598
ADDW4  = 000000           598
ADDW5  - 000000           598
ADDW6  = 000000           598
ADDW7  = 000000           598
ADDW8  = 000000           598
ADDW9  = 000000           598
ADEVCT=  000000           598     604
ADEVM  - 000000           598
ADRNG    067415         10301   11814#
AENV   = 000000           598     609
AENVM  = 000000           598     610
AFATAL=  000000           598     601
AMADR1=  000000           598
AMADR2=  000000           598
AMADR3=  000000           598
```

```
AMADR4= 000000           598
AMAMS1= 000000           598
AMAMS2= 000000           598
AMAMS3= 000000           598
AMAMS4= 000000           598
AMSGAD= 000000           598      606
AMSGLG= 000000           598      607
AMSGTY= 000000           598      600
AMTYP1= 000000           598
AMTYP2= 000000           598
AMTYP3= 000000           598
AMTYP4= 000000           598
APASS = )00000           598      603
APRIOR= .00000           598
APTCSU= 000040          9527#    9745
APTENV= 000001          9444     9483     9525#    9738
APTSIZ= 000200          1331     9524#
APTSPO= 000100          9485     9526#    9740
ASLOC   047242          8809*    8815     8824     8830     8837#
ASRBCB  047076          8800#
ASWREG= 000000           598      611
ATESTN= 000000           598      602
AUNIT = 000000           598      605
AUSWR = 000000           598      612
AVECT1= 000000           598
AVECT2= 000000           598
AVMBL   042222          7724*    7730#    7750
BACKAD  055662         10271*   10280    10281#
BB    = 000014          3174#
BBADR1  015136          3204     3240     3242     3257     3278*    3279*    3295#    3316     3318     3344     3345     3348     3349
                        3351     3354    14307
BBADR2  015142          3297#    3348*    3349*   14310
BBADR3  015146          3299#    3350*    3351*    3352*    3353*    3354*    3355*   14310
BBCNT1  015156          3232*    3233*    3307#    3358*    3359*   14310
BBDONE  015446          3289     3373#
BBERR1  015166          3235     3313#    3370
BBERR2  015224          3324     3333#
BBERR3  015240          3334     3337#
BBERR4  015424          3361     3368#
BBERR5  015426          3367     3369#
BBFLG1  015152          3187*    3275     3285     3288     3290*    3302#    3338
BBFLG2  015154          3231*    3282     3304#    3357*
BBGM    015164          3189*    3196     3292*    3311#
BBGS    015162          3188*    3194     3200     3291*    3310#
BBHIAD  014452          3185#
BBLOAD  014450          3184#    3240     3242
BB0     014430          3175#    3192
BB1     014474          3191#    3293
BB2     014512          3194#    3198
BB3     014552          3204#
BB4     014720          3237#    3280
BB5     014756          3248     3253     3257#
BB6     015040          3273     3278#    3371
BB7     015054          3251     3282#
BB8     015102          3283     3288#
BIT0  - 000001           137#     429     1429     1430     1437     1755     5421     5575    10171
```

```
BIT00 = 000001            127#     137
BIT01 = 000002            126#     136
BIT02 = 000004            125#     135
BIT03 = 000010            124#     134
BIT04 = 000020            123#     133
BIT05 = 000040            122#     132
BIT06 = 000100            121#     131
BIT07 = 000200            120#     130     1379
BIT08 = 000400            119#     129     9376
BIT09 = 001000            118#     128     9386     9455
BIT1  = 000002            136#     428
BIT10 = 002000            117#     421     9432
BIT11 = 004000            116#     420     1432     1433     1435     1639     9393     11048    11209
BIT12 = 010000            115#     419     11044    11055    11207    11668
BIT13 = 020000            114#     418     9439
BIT14 = 040000            113#     417     1424     1425     1427     1618     8370     9362     11046    11057    11214    11670    11676
BIT15 = 100000            112#     416      431     1438     1439     1441     1443     1444     1446     8350     8373     8377     10174
BIT2  = 000004            135#     427      433      434      435
BIT3  = 000010            134#     426      433      434      435
BIT4  = 000020            133#     425      434     8910     8914
BIT5  = 000040            132#     424      433     8912     8914     11024    11190    11393    11650
BIT6  = 000100            131#     1508    10411    10975    11042    11134    11301    11399    11475    11605
BIT7  = 000200            130#    10841    10842    10849
BIT8  = 000400            129#     423     1458     11040    11205    11666
BIT9  = 001000            128#     422     11050
BOTPRG= 126310           2274     2674     2710     2715     2730     2799     2814     2838     3205     14424#
BOTTOM  120310           1499    10383    14423#    14424
BPTVEC- 000014            144#
BYP   = 100000            431#    9049     9055     9056     9062     9063     9085     9091     9092     9098     9099     9120     9126
                         9127     9133     9134     9235     9266
CACHVE= 000114            152#    1522*    1573*    1655*    1656*    1681*    1682*    1706*    1753*    1849*    1978*    2080*    2215*
                         2299*    2411*    2504*    2613*    2695*    2767*    2898*    3191*    3235*    3361*    3370*    3419*    3513*
                         3538*    3616*    3687*    3738*    3917*    3927*    3983*    4036*    4216*    4226*    4268*    4401*    4540*
                         4699*    4871*    4877*    4935*    5119*    5236*    5243*    5363*    5371*    5415*  · 5425*    5468*    5569*
                         5579*    5622*    5713*    5799*    5969*    6308*    10233*   10240*   10269*
CALRH4  104430           7289     7309     7329     7347     7835     10036#
CALRK5= 104431           7458     7478     7498     7516     10037#   10819
CALRP4= 104427           7125     7145     7165     7183     7882     10035#   10790
CALRS4= 104426           6957     6977     6997     7015     7859     10034#   10762
CALUBE= 104432           7604     7621     10038#
CBP     046532           8690#    8699
CBPA    046544           8693#    8774     8780
CC    = 000020           4263#
CCDONE  022372           4321     4365     4383#
CCERR1  022100           4268     4328#
CCERR2  022126           4332     4334#
CCERR3  022312           4281     4292     4304     4316     4368#
CCTMP1  022072           4323#
CCTMP2  022074           4272     4324#
CC1     021662           4275#
CC10    022024           4305     4309#    4364
CC11    022054           4316#    4319
CC12    022062           4315     4318#
CC13    022066           4317     4321#
CC2     021704           4281#    4284
CC3     021712           4280     4283#
```

```
CC4     021720         4282     4285#    4356
CC5     021746         4292#    4295
CC6     021754         4291     4294#
CC7     021762         4293     4297#    4360
CC8     022012         4304#    4307
CC9     022020         4303     4306#
CHAINQ  056106         9330    10382#
CISP    001751          625#
CKSWR = 104407         9361     9428     9454    10017#
CLEAN   055574        10023    10267#
CNRNG   067625        10315    11841#
CONCMS  066370        10381    11694#
CONFLG  056010        10311    10337#
CONFL2  056024        10343#
CONTRL= 177746          161#     437#    1421     1451*    1452     1520*    1880*    2112*    2328*    2527*    2697*    2769*    2973*
                       2975*    3194*    3196*    3202*    3459*    3704*    3706*    3712*    3805*    3916     3920*    3926*    4002*
                       4004*    4010*    4106*    4215     4219*    4225*    4270*    4403*    4564*    4724*    4962*    4967*    4973*
                       5040*    5053*    5129*    5131*    5136*    5153*    5247*    5249*    5259*    5272*    5424*    5470*    5578*
                       5624*    5740*    5742*    5750*    5763*    5971*    6006*    6009*    6012*    6078*    6080*    6082*    6084*
                       6086*    6088*    6090*    6092*    6097*    6099*    6101*    6103*    6105*    6107*    6109*    6111*    6115*
                       6117*    6134*    6136*    6154*    6156*    6173*    6175*    6199*    6280*    6284*    6334*    6337*    6340*
                       7925*    7927*    7933*    7940*    8024*    8025     8027     8030*    8031     8033     8036*    8038     8041*
                       8042     8044     8048     8051     8054*    8055     8057     8059     8070*    8071     8073     8076     8078*
                       8079     8081     8084*    8085     8087     8099*    8102     8104     8107     8110     8125*    8126     8128*
                       8129     8131     8134     8136*    8137     8139     8142     8154*    8155     8157*    8158*    8159     8161
                       8165     8167*    8168     8170*    8171     8173     8177     8202*    8203     8211*    8213*    8218*    8219*
                       8222*    8223*    8224*    8225     8229     8232*    8237*    8243     8258     8261*    8262     8264*    8269*
                       8279     8319*    8320     8326*    8328*    8336*    8341*    8348     8405*    8406     8414*    8416*    8421*
                       8422*    8425*    8426*    8427*    8428     8432     8435*    8440*    8446     8461     8464*    8465     8467*
                       8472*    8482     8521*    8522     8527*    8529*    8539*    8543*    8544*    8549*    8557     8579*    8582*
                       8592*    8597*    8603     8645*    8646     8650*    8651*    8695*    8696     8701*    8703*    8712*    8715*
                       8717*    8725     8744*    8751     8802*    8804*    8807*    8813     8828     8855*    8856     8859*    8862*
                       8867*    8869*    8873*    8879*    8880     8893     8895     8900     8902     8917     8928     8932     8980*
                       8981     8984*    8986*    8988*    8989*    8997     9004     9007     9013*    9014     9016     9033*    9164*
                       9165     9170*    9172*    9176*    9179*    9183*    9220     9241     9271     9273    10276*
CPSPUR  055412         1509     1558     1599     1705     2300    10225#   10270
CPUERR= 177766          173#    1708*    9467*   10227    10278*   10661*   10672*   10690*   10700*   10710*
CR    = 000015           49#    9784     9794    12070
CRLF  = 000200           50#    1355     1362     9755     9794    11694    11696    11703    11713    11716    11723    11733    11743
                      11746    11756    11762    11765    11770    11777    11782    11789    11799    11802    11809    11814    11828
                      11841    11853    11867    11880    11890    11903    11908    11916    11922    11928    11933    11939    11945
                      11958    11970    11975    11981    11991    11997    12002    12008    12019    12031    12053    12059    12061
                      12063    12068    12102    12125    12139    12145    12154    12163    12174    12206    12214    12224    12279
                      12306    12327    12347    12369    12382    12395    12409    12436    12465    12487    12493    12502    12514
                      12539    12562    12586    12602    12627    12819    12828    13012    13028    13045    13054    13072    13079
                      13092    13114    13121    13134    13156    13163    13176    13198    13205    13218    13240    13247    13260
                      13280    13287    13300    13321    13328    13342    13362    13369    13382    13393    13450    13456    13463
                      13475    13506    13530    13547
CVSPE   047256         8850#
CVSPEA  047270         8853#    8950     8958
CYCNT   036232         6818     6847#
DD    = 000024         4866#
DDCNTR  025204         4881*    4897     4904     4911*    4918#
DDDONE  025206         4901     4906     4920#
DDISP = 177570           43#     537     1319
DDPD    025054         4880     4887#
```

```
DDPER    025152          4877    4908#
DDPER1   025176          4909    4914#
DDPU1    025112          4896#
DDPU2    025136          4898    4903#
DDPV     025066          4888    4890#
DDTMP    025202          4879*   4891     4896     4916#
DD1      025020          4874    4877#
DF1      114521           648   13994#
DF10   = 114533           669   14012#
DF100    115002          1053    1060     1067    14143#
DF103  = 114774           943   14145#
DF104  = 114774           950   14146#
DF105  = 114774           957   14147#
DF106  = 114774           964   14148#
DF107    115010           971   14149#
DF11     114543           672   14014#
DF110  = 114774           978   14152#
DF111  = 114774           985   14153#
DF112  = 114774           992   14154#
DF113  = 114776           999   14155#
DF115  = 114774          1011   14156#
DF12     114554           675   14018#
DF123    115015          1042    1047     1072     1078     1085     1091    14157#
DF13     114562           678   14021#
DF136    115020          1099   14158#   14160
DF137  = 115020          1102   14160#
DF14     114567           681   14024#
DF140    115024          1105   14162#   14165    14167    14169    14171    14173    14175    14177
DF141  = 115024          1108   14165#
DF142  = 115024          1111   14167#
DF143  = 115024          1114   14169#
DF144  = 115024          1117   14171#
DF145  = 115024          1120   14173#
DF146  = 115024          1123   14175#
DF147  = 115024          1126   14177#
DF15     114574           684   14027#
DF150    115030          1129   14179#
DF151    115034          1132   14182#   14184    14185
DF152  = 115034          1135   14184#
DF153  = 115034          1138   14185#
DF154    115037          1141   14187#   14190    14191    14192    14193    14194
DF155  = 115037          1144   14190#
DF156  = 115037          1147   14191#
DF157  = 115037         14192#
DF16     114576           687   14029#
DF160  = 115037          1153   14193#
DF161  = 115037          1156   14194#
DF17     114604           690   14032#   14041
DF2      114525           651   13997#   14000    14002
DF20   = 114604           693   14041#
DF21     114634           696   14043#
DF22     114662           699   14052#   14057    14067    14071
DF23     114665           702   14054#   14059    14069    14073
DF24   = 114662           705   14057#
DF25   = 114665           708   14059#
DF26     114672           711   14061#
```

```
DF27      114676           714    14064#
DF3     = 114525           654    14000#
DF30    = 114662           717    14067#
DF31    = 114665           720    14069#
DF32    = 114662           723    14071#
DF33    = 114665           726    14073#
DF34      114703           729    14075#  14078
DF35    = 114703           732    14078#
DF36      114710           735    14080#
DF37      114714           738    14083#
DF4     = 114525           657    14002#
DF40    = 114726           741    14096#
DF41      114722           744    14087#  14090
DF42    = 114722           747    14090#
DF43      114726           750    14092#  14096
DF44      114736           753    14098#  14104   14113   14115
DF45    = 114736           756    14104#
DF46      114755           759    14106#  14111   14117   14119
DF47    = 114755           762    14111#
DF5       114533           660    14004#  14008   14010   14012
DF50    = 114736           765    14113#
DF51    = 114736           768    14115#
DF52    = 114755           771    14117#
DF53    = 114755           774    14119#
DF54      114771           777    14121#
DF55      114774          1160     1164    1168    1172    1176    1180    1184    1188    1192    1196    1200    1204   14122#
                         14123    14124   14125   14126   14127   14128   14129   14130   14145   14146   14147   14148   14152
                         14153    14154   14156
DF56    = 114774         14123#
DF57    - 114774           802    14124#
DF6     = 114533           663    14008#
DF60    = 114774         14125#
DF61    = 114774           810      817   14126#
DF62    = 114774           825    14127#
DF63    = 114774           833    14128#
DF64    = 114774           841    14129#
DF65    = 114774           849    14130#
DF66      114776           857    14131#  14133   14134   14135   14136   14137   14138   14139   14140   14141   14155
DF67    = 114776           865    14133#
DF7     = 114533           666    14010#
DF70    = 114776           873    14134#
DF71    = 114776           880    14135#
DF72    = 114776           887    14136#
DF73    = 114776           894    14137#
DF74    = 114776           901    14138#
DF75    = 114776           908    14139#
DF76    = 114776           915    14140#
DF77    = 114776           922    14141#
DH1       111261           648    13577#
DH10    = 111435           669    13613#
DH107     113367           969    13847#
DH11      111467           672    13615#
DH111     113441           983    13855#
DH115     113456          1009    13858#
DH12      111543           675    13624#
DH123     113475          1040     1045   13862#
```

```
DH125     113524        1051    1058     1065     13866#
DH13      111636         678   13636#
DH130     113613        1070    1076    13876#
DH132     113642        1083    1089    13880#
DH136     113671        1099   13886#   13894
DH137  =  113671        1102   13894#
DH14      111711         681   13645#
DH140     113736        1105   13896#   13903    13905    13907    13909    13911    13913    13915
DH141  =  113736        1108   13903#
DH142  =  113736        1111   13905#
DH143  =  113736        1114   13907#
DH144  =  113736        1117   13909#
DH145  =  113736        1120   13911#
DH146  =  113736        1123   13913#
DH147  =  113736        1126   13915#
DH15      112004         684   13657#
DH150     114001        1129   13917#
DH151     114065        1132   13927#
DH152     114134        1135   13935#
DH153     114203        1138   13944#
DH154     114265        1141   13955#
DH155     114323        1144   13962#
DH156     114361        1147   13969#
DH16      112030         687   13662#
DH160     114417        1153   13977#
DH161     114445        1156   13982#
DH162     114475        1160    1164     1168     1172     1176     1180     1184     1188     1192     1196     1200     1204    13987#
DH17      112116         690   13673#   13683
DH2       111334         651   13586#   13599    13601
DH20   =  112116         693   13683#
DH21      112176         696   13685#
DH22      112251         699   13694#   13713    13737    13741
DH23      112320         702   13702#   13715    13739    13743
DH24   =  112251         705   13713#
DH25   =  112320         708   13715#
DH26      112410         711   13717#
DH27      112455         714   13725#
DH3    =  111334         654   13599#
DH30   =  112251         717   13737#
DH31   =  112320         720   13739#
DH32   =  112251         723   13741#
DH33   =  112320         726   13743#
DH34      112547         729   13745#   13755
DH35   =  112547         732   13755#
DH36      112627         735   13757#
DH37      112674         738   13765#
DH4    =  111334         657   13601#
DH40   =  113026         741   13795#
DH41      112761         744   13777#   13785
DH42   =  112761         747   13785#
DH43      113026         750   13787#   13795
DH44      113075         753   13797#   13809    13818    13820
DH45   =  113075         756   13809#
DH46      113170         759   13811#   13816    13822    13824
DH47   =  113170         762   13816#
DH5       111435         660   13603#   13609    13611    13613
```

```
DH50    = 113075            765    13818#
DH51    = 113075            768    13820#
DH52    = 113170            771    13822#
DH53    = 113170            774    13824#
DH54      113214            777    13826#
DH55      113255            783      792      800      808      815      823      831      839      847      941      948      955      962
                            976      990    13832#
DH6     = 111435            663    13609#
DH66      113271            855      863      871      885      892      899      906      913      920      997    13835#
DH7     = 111435            666    13611#
DH71      113335            878    13842#
DISPLA    001542            537#    1319*    1327*    9408*    9431*
DISPRE    000174            466#    1327
DMMA    = 004000            420#
DRH4T1    040562           7233*    7247     7250*    7260     7388#
DRH4T2    040564           7234*    7248     7251*    7261     7389#
DRK5T1    041374           7398*    7412     7415*    7425     7557#
DRK5T2    041376           7399*    7413     7416*    7426     7558#
DRP4T1    037770           7065*    7079     7082*    7092     7224#
DRP4T2    037772           7066*    7080     7083*    7093     7225#
DRRH4     037774           6864     7232#
DRRK5     040566           6865     7397#
DRRP4     037162           6863     7064#
DRRS4     036350           6862     6896#
DRS4T1    037156           6897*    6911     6914*    6924     7056#
DRS4T2    037160           6898*    6912     6915*    6925     7057#
DRUBE     041400           6866     7564#
DSWR    = 177570             42#     536     1318
DT      = 000001            429#    8989
DT1       115044            648    14201#
DT10    = 115074            669    14220#
DT100     116136          14364#
DT107     116154            970    14367#
DT11      115116            672    14222#
DT12      115142            675    14227#
DT123     116170           1041     1046     1071     1077     1084     1090    14370#
DT125     116200           1052     1059     1066    14372#
DT13      115160            678    14231#
DT136     116216           1099    14376#   14379
DT137   = 116216           1102    14379#
DT14      115174            681    14234#
DT140     116230           1105    14381#   14384    14386    14388    14390    14392    14394    14396
DT141   = 116230           1108    14384#
DT142   = 116230           1111    14386#
DT143   = 116230           1114    14388#
DT144   = 116230           1117    14390#
DT145   = 116230           1120    14392#
DT146   = 116230           1123    14394#
DT147   = 116230           1126    14396#
DT15      115210            684    14237#
DT150     116242           1129    14398#
DT151     116254           1132    14401#   14404    14405    14410    14411    14412
DT152   = 116254           1135    14404#
DT153   = 116254           1138    14405#
DT154     116264           1141    14406#   14408    14409
DT155   = 116264           1144    14408#
```

```
DT156 = 116264           1147   14409#
DT157 = 116254          14410#
DT16    115216            687   14239#
DT160 = 116254           1153   14411#
DT161 = 116254           1156   14412#
DT162   116276           1160    1164     1168     1172     1176     1180     1184     1188     1192     1196     1200     1204     14413#
DT17    115234            690   14243#
DT2     115056            651   14204#   14208    14210
DT20    115316            693   14253#
DT21    115400            696   14263#
DT22    115456            699   14272#
DT23    115466            702   14275#
DT24    115502            705   14278#
DT25    115512            708   14281#
DT26    115526            711   14284#
DT27    115540            714   14287#
DT3   = 115056            654   14208#
DT30    115554            717   14290#
DT31    115564            720   14293#
DT32    115600            723   14296#
DT33    115610            726   14299#
DT34    115624            729   14302#   14305
DT35  = 115624            732   14305#
DT36    115640            735   14307#
DT37    115652            738   14310#
DT4   = 115056            657   14210#
DT40  = 115700            741   14323#
DT41    115666            744   14314#   14317
DT42  = 115666            747   14317#
DT43    115700            750   14319#   14323
DT44    115722            753   14325#   14332
DT45  = 115722            756   14332#
DT46    115762            759   14334#   14340
DT47  = 115762            762   14340#
DT5     115074            660   14212#   14216    14218    14220
DT50    116014            765   14342#   14349
DT51  = 116014            768   14349#
DT52    116054            771   14351#   14357
DT53  = 116054            774   14357#
DT54    116106            777   14359#
DT55    116116            784     793      801      809      816      824      832      840      848      942      949      956      963
                          977     984      991     1010    14361#

DT6   = 115074            663   14216#
DT66    116124            856     864      872      879      886      893      900      907      914      921      998    14362#
DT7   = 115074            666   14218#
DUBET1  041746           7564*   7583     7586*    7590     7655#
DUBET2  041750           7565*   7584     7587*    7656#
DUT   = 000002            428#
EE    = 000022           4534#
EEDONE  024056           4617    4661     4679#
EEERR1  023564           4540    4624#
EEERR2  023612           4628    4630#
EEERR3  023776           4576    4588     4600     4612     4664#
EETMP1  023556           4619#
EETMP2  023560           4566    4620#
EE1     023346           4570#
```

```
EE10     023510           4601     4605#    4660
EE11     023540           4612#    4615
EE12     023546           4611     4614#
EE13     023552           4613     4617#
EE2      023370           4576#    4579
EE3      023376           4575     4578#
EE4      023404           4577     4581#    4652
EE5      023432           4588#    4591
EE6      023440           4587     4590#
EE7      023446           4589     4593#    4656
EE8      023476           4600#    4603
EE9      023504           4599     4602#
EMTVEC=  000030            147#    1302*    1303*
EM1      072022            648    12083#
EM10     072762            669    12174#
EM100    101464          12837#
EM103    101524            940    12843#
EM104    101605            947    12852#
EM105    101711            954    12864#
EM106    101765            961    12872#
EM107    102055            968    12882#
EM11     073053            672    12186#
EM110    102174            975    12896#
EM111    102222            982    12900#
EM112    102304            989    12909#
EM113    102345            996    12915#
EM115    102403           1008    12921#
EM12     073141            675    12197#
EM123    102520           1039    12936#
EM124    102566           1044    12943#
EM125    102630           1050    12949#
EM126    102670           1056    12955#
EM127    103021           1063    12970#
EM13     073266            678    12214#
EM130    103150           1069    12986#
EM131    103216           1075    12993#
EM132    103260           1082    12999#
EM133    103326           1088    13006#
EM136    103370           1099    13012#
EM137    103605           1102    13037#
EM14     073346            681    12224#
EM140    104023           1105    13063#
EM141    104364           1108    13105#
EM142    104724           1111    13147#
EM143    105266           1114    13189#
EM144    105627           1117    13231#
EM145    106161           1120    13271#
EM146    106512           1123    13312#
EM147    107045           1126    13353#
EM15     073405            684    12231#
EM150    107377           1129    13393#
EM151    107463           1132    13403#   13413    13414
EM152 =  107463           1135    13413#
EM153 =  107463           1138    13414#
EM154    107544           1141    13416#
EM155    107576           1144    13422#
```

```
EM156     107630          1147    13428#
EM16      073455           687    12240#
EM160     107675          1153    13437#
EM161     107727          1156    13443#
EM162     107775          1160    13450#
EM163     110205          1164    13475#
EM164     110277          1168    13486#
EM165     110356          1172    13494#
EM166     110375          1176    13497#
EM167     110461          1180    13506#
EM17      073531           690    12249#   12259
EM170     110565          1184    13518#
EM171     110630          1188    13524#
EM172     110674          1192    13530#
EM173     110760          1196    13540#
EM174     111146          1200    13561#
EM175     111211          1204    13567#
EM2       072107           651    12093#
EM20    = 073531           693    12259#
EM21      073612           696    12261#
EM22      073676           699    12271#   12315     12358     12378
EM23      074112           702    12297#   12317
EM24    = 073676           705    12315#
EM25    = 074112           708    12317#
EM26      074246           711    12319#
EM27      074413           714    12338#
EM3       072301           654    12117#
EM30    = 073676           717    12358#
EM31      074560           720    12360#   12380
EM32    = 073676           723    12378#
EM33    = 074560           726    12380#
EM34      074717           729    12382#
EM35      075023           732    12395#
EM36      075132           735    12409#
EM37      075264           738    12426#
EM4       072415           657    12131#
EM40      075346           741    12436#
EM41      075521           744    12457#
EM42      075705           747    12479#
EM43      076160           750    12514#
EM435     111026         13547#
EM44      076306           753    12531#   12610
EM45      076502           756    12554#   12612
EM46      076701           759    12578#   12614
EM47      077022           762    12594#   12616
EM5       072530           660    12145#
EM50    = 076306           765    12610#
EM51    = 076502           768    12612#
EM52    = 076701           771    12614#
EM53    = 077022           774    12616#
EM54      077146           777    12618#
EM55      077321           782    12637#
EM56      077352           791    12642#
EM57      077420           799    12649#
EM6       072610           663    12154#
EM60      077461           807    12655#
```

```
EM61     C77520           81     12661#
EM62     077607          822     12671#
EM63     077641          830     12676#
EM64     077717          838     12684#
EM65     100000          846     12693#
EM66     100066          854     '12702#
EM67     100176          862     12714#
EM7      072670          666     12163#
EM70     100311          870     12727#
EM71     100425          877     12740#
EM72     100473          884     12747#
EM724    071730         1740     12070#
EM73     100622          891     12763#
EM74     100732          898     12776#
EM75     101034          905     12787#
EM76     101150          912     12800#
EM77     101261          919     12813#
ENDKB    005224         1461      1466#
ERRNG    067525        10308     11828#
ERRVEC=  000004          140#     1316     1317*    1328*    1523*    1558*    1572*    1599*    1654*    1705*    1730*    1775*    2300*
                        9367      9368*    9370*    9373*   10113    10114    10118*   10123*   10135*   10140*   10150*   10160*   10161*
                       10270*
ERTYPE   056354         3286      5312     9441    10258    10467#
ERT1     056466        10497#    10562
ERT2     056700        10501     10506    10516    10524    10530    10542    10552    10556#
ERT3     056704        10536     10559#
ERT4     056714        10495     10561    10564#
ERT5     056716        10475     10565#
FCAC   = 000400          423#     8099     8128     8136     8158     8170     8224     8427     8549     8650     8651
FF     = 000021         4396#
FFDONE   023170         4456      4500     4518#
FFERR1   022676         4401      4463#
FFERR2   022724         4467      4469#
FFERR3   023110         4415      4427     4439     4451     4503#
FFTMP1   022670         4458#
FFTMP2   022672         4405      4459#
FF1      022446         4409#
FF10     022620         4440      4444#    4499
FF11     022650         4451#     4454
FF12     022656         4450      4453#
FF13     022664         4452      4456#
FF2      022472         4415#     4418
FF3      022500         4414      4417#
FF4      022510         4416      4420#    4491
FF5      022536         4427#     4430
FF6      022544         4426      4429#
FF7      022554         4428      4432#    4495
FF8      022604         4439#     4442
FF9      022612         4438      4441#
FVPE   = 002000          421#     8873     8879     8893     8989
GETBUF   042232         6904      7072     7240     7405     7569     7738#
GETMP1   042230         7733#     7739*    7760
GGFLG1   027024         5122*     5146     5170     5178     5193     5202     5213*    5217#
GGGM     027030         5124*     5131     5151     5212*    5220#
GGGS     027026         5123*     5129     5134     5211*    5219#
GG1      026354         5122#
```

```
GG10    026736            5192    5198#
GG11    026772            5201    5208#
GG12    027032            5214    5222#
GG2     026376            5127#   5215
GG3     026406            5129#   5133
GG4     026464            5139    5140#   5150
GG5     026556            5157    5158#   5161
GG6     026604            5165    5166#   5185
GG7     026634            5169    5175#
GG8     026664            5176    5184#
GG9     026706            5188    5189#   5209
GNS   = ****** U           465    1354   10009   10010   10011   10012   10013   10015   10017   10018   10019   10020   10021
                         10023   10024   10025   10026   10027   10028   10029   10030   10031   10032   10034   10035   10036
                         10037   10038
GTBIHI  042136            7716#   7718*   7719
GTBILO  042134            7715#   7717*   7720
GTBINT  042116            6784    7709#
GTMSIZ  042220            7722*   7729#
GTRNH   042226            7726*   7732#   7744    7748*
GTRNL   042224            7725*   7731#   7743    7746*
GTSWR = 104406            1349   10015#  10409
G1GOV   045016            8315#   8324
G1GOVA  045022            8317#   8372
G1GOVB  045032            8319#   8378
HH    = 000023            4691#
HHDONE  024756            4777    4821    4839#
HHERR1  024464            4699    4784#
HHERR2  024512            4788    4790#
HHERR3  024676            4736    4748    4760    4772    4824#
HHTMP1  024456            4779#
HHTMP2  024460            4726    4780#
HH1     024234            4730#
HH10    024406            4761    4765#   4820
HH11    024436            4772#   4775
HH12    024444            4771    4774#
HH13    024452            4773    4777#
HH2     024260            4736#   4739
HH3     024266            4735    4738#
HH4     024276            4737    4741#   4812
HH5     024324            4748#   4751
HH6     024332            4747    4750#
HH7     024342            4749    4753#   4816
HH8     024372            4760#   4763
HH9     024400            4759    4762#
HIADRS= 177742             159#   2016    2250    2450    2649    3343    3912    4211    4339    4474    4635    4795    5358
                          5920   10243
HIAFLG  056004           10297   10335#
HIAFL2  056020           10341#
HIMFLG  056014           10325   10339#
HIMFL2  056030           10345#
HITMIS= 177752             163#    438#   3047    3067    3272    3526    3789    3860    4088    4161    4978    4994    5010
                          5026    5144    5168    5191    5263    5287    5431    5477    5585    5631    5754    6018    6032
                          6047    6061    6121    6140    6160    6179    6346    6360    6375    6389    6422    6439    6458
                          6475    6494    6511    6530    6547    6572    6595    6618    6641    7806    8241    8277    8346
                          8444    8480    8551    8601    8655    8657    8722    8748    8810    8825    8995    9218    9239
HMRNG   070035           10329   11867#
```

```
HT     = 000011          47#     9753    9794
IIA    = 000031        5410#
IIAR1    030350        5425     5518#
IIAR2    030410        5468     5528#
IIAT1    030346        5426     5464     5472     5510     5516#    5522     5532
IIA1     030046        5419#    5541
IIA2     030072        5422     5424#
IIA3     030172        5461#
IIA4     030212        5468#    5469     5527
IIA5     030324        5507#
IIA6     030344        5514#
IIA7     030444        5423     5514     5538#
IIA8     030462        5539     5543#
IIB    = 000032        5564#
IIBR1    031024        5579     5672#
IIBR2    031064        5622     5682#
IIBT1    031022        5580     5618     5626     5664     5670#    5676     5686
IIB1     030520        5573#    5695
IIB2     030544        5576     5578#
IIB3     030646        5615#
IIB4     030666        5622#    5623     5681
IIB5     031000        5661#
IIB6     031020        5668#
IIB7     031120        5577     5668     5692#
IIB8     031136        5693     5697#
INDONE   042350        6889     7767#
INTMP1   036230        6809*    6830     6844#
INT0     036022        6784#
INT1     036040        6791#
INT2     036064        6797#    6836
INT3     036112        6807#    6814
INT4     036130        6808     6812#
INT5     036144        6811     6818#
INT6     036212        6800     6816     6832#
IOTVEC=  000020         145#    1300*    1301*
IVFC     045766        8516#    8525     8575
IVFCA    046000        8519#    8625     8632
IVSS   = 040000         417#    8030     8031     8036     8070     8157     8544
JJCNT    026274        4938*    5072*    5074     5080     5086     5092#
JJPAT1   026276        4949*    4959     5042     5071     5076*    5082*    5088*    5094#   14212
JJPAT2   026300        4947     4954*    4960     5048     5058     5059     5076     5077*    5095#   14212
JJPAT3   026302        4956*    4963     4964     4965     4984     5000     5082     5083*    5096#   14212
JJPAT4   026304        4957*    4968     4969     4970     5016     5032     5054     5088     5089*    5097#   14212
JJPAT5   026306        5057*    5060     5063*    5065     5071*    5077     5083     5089     5098#
JJTMP1   026310        5100#
JJTMP2   026312        4939     5101#
JJ1      025344        4958#    5069     5078     5084     5090
JJ10     026012        5038     5039#
JJ11     026046        5046     5047#
JJ12     026066        5049     5053#
JJ13     026122        5055     5063#
JJ14     026136        5061     5067#
JJ15     026214        5075     5080#
JJ16     026244        5081     5086#
JJ17     026322        5087     5105#
JJ2      025452        4975#
```

```
JJ3      025512        4979    4984#
JJ4      025542        4990    4991#
JJ5      025602        4995    5000#
JJ6      025632        5006    5007#
JJ7      025672        5011    5016#
JJ8      025722        5022    5023#
JJ9      025762        5027    5032#
KBTST    004750        1413#
KB11CM   001750         624#   1413*    1460*    1472     8009     9043     9079     9114     9156
KB11E    001746         622#   1414*    1418*    1456     1458*    1470     1478     9045     9081     9116
KB11EM   001747         623#   8011     9158
KDPAR0=  172360         315#
KDPAR1-  172362         316#
KDPAR2=  172364         317#
KDPAR3=  172366         318#
KDPAR4=  172370         319#
KDPAR5=  172372         320#
KDPAR6=  172374         321#
KDPAR7=  172376         322#
KDPDR0=  172320         293#
KDPDR1=  172322         294#
KDPDR2=  172324         295#
KDPDR3=  172326         296#
KDPDR4=  172330         297#
KDPDR5=  172332         298#
KDPDR6=  172334         299#
KDPDR7=  172336         300#   9070
KERSTK=  001100          33#
KIPAR0=  172340         304#   1855     2088     2304     2507     2782     2906     3213     3422     3715     4013     4542     4702
                       6766    9185*    10087    10127    10599
KIPAR1=  1/2342         305#   9187*
KIPAR2=  172344         306#   9189*
KIPAR3=  172346         307#   9191*
KIPAR4=  172350         308#   9162
KIPAR5=  172352         309#
KIPAR6=  172354         310#   1611*    1747*    1966*    2203*    2398*    2600*    2805*    2820*    2844*    3041*    3061*    3264*
                       3511*   3782*    3855*    4082*    4156*    7690*
KIPAR7=  172356         311#   9193*
KIPDR0=  172300         282#   1438*    1439     1441*    1857     2090     2306     2509     2784     2908     3215     3424     3717
                       4015    4544     4704     6768     9047     9186*    9298
KIPDR1=  172302         283#   9188*
KIPDR2=  172304         284#   9190*
KIPDR3=  172306         285#   9192*
KIPDR4=  172310         286#   9163
KIPDR5=  172312         287#
KIPDR6=  172314         288#   1612*    1685*    1725*
KIPDR7=  172316         289#   9194*
KKCNT1   033674        6002*   6220*    6222     6229     6236     6247#
KKERR1   033722        6131    6260#
KKERR2   033734        6151    6264#
KKERR3   033754        6170    6269#
KKERR4   033770        6190    6273#
KKERR5   034012        6262    6267     6271     6276     6278#
KKFLG1   033672        5977*   6072     6192     6194*    6198*    6244#
KKPAT1   033676        5972*   6079     6110     6116     6219     6225*    6232*    6239*    6250#   14268
KKPAT2   033700        5973*   5998     6083     6106     6135     6205     6206     6225     6226*    6251#   14268
```

```
KKPAT3   033702           5974*   6087    6102    6155    6232    6233*   6252#   14268
KKPAT4   033704           5975*   6091    6098    6174    6201    6239    6240*   6253#   14268
KKPAT5   033706           6204*   6207    6210*   6212    6219*   6226    6233    6240    6254#
KKTMP1   033710           6256#
KKTMP2   033712           5983    6257#
KK1      032330           5971#
KK10     033436           6189    6192#
KK11     033456           6193    6198#
KK12     033520           6202    6210#
KK13     033534           6208    6214#
KK14     033544           6215    6218#
KK15     033612           6223    6229#
KK16     033642           6230    6236#
KK17     034040           6237    6287#
KK2      032370           5983#
KK3      032474           6004    6005#   6195    6216    6227    6234    6241
KK4      033022           6075    6095#
KK5      033132           6094    6114#
KK6      033210           6130    6133#
KK7      033274           6150    6153#
KK8      033352           6169    6172#
KT       050102           7783    8013    9034#
LF     = 000012             48#   9788    9794    12070   12493
LKS    = 177546             44#
LKVEC  = 00010C            151#
LLCNT1   027532           5280*   5300*   5311    5328#
LLERR1   027540           5334#   5371
LLERR2   027744           5294    5374#
LLERR3   027724           5363    5370#
LLFLG1   027520           5239*   5265    5289    5308    5316*   5320#   5336    5354    5380
LLFLG2   027522           5281*   5301*   5305    5321#
LLFLG4   027524           5283*   5298    5323#   5347*   5374*
LLGM     027530           5241*   5247    5270    5314*   5326#
LLGS     027526           5240*   5249    5257    5315*   5325#
LLMER    027534           5330#   5334*   5339    5360
LLTMP1   027536           5332#   5335*   5338*   5339
LL1      027064           5239#
LL10     030012           5317    5388#
LL2      027106           5242#   5244    5318
LL3      027132           5247#   5252
LL4      027302           5276#
LL5      027332           5282    5283#   5303
LL6      027370           5288    5292#
LL7      027400           5293    5296#   5372    5387
LL8      027420           5299    5302#
LL9      027472           5306    5314#
LOADRS=  177740            158#   1991    1993    2015    2226    2228    2249    2425    2427    2449    2624    2626    2648
                          3316    3318    3342    3364    3624    3626    3911    3918    4210    4217    4331    4338    4466
                          4473    4627    4634    4787    4794    5343    5357    5522    5532    5676    5686    5919    10237
                          10242
LOAFLG   056002           10295   10334#
LOAFL2   056016           10340#
LOC    = 030770           1530#   1531#   1532#   1533    1578#   1579#   1580#   1581    1662#   1663#   1664#   1665    1689#
                          1690#   1691#   1692    1757#   1758#   1759#   1760    5448#   5449#   5450#   5451    5497#   5498#
                          5499#   5500    5602#   5603#   5604#   5605    5651#   5652#   5653#   5654
LOOP     005306           1493#   9340
```

```
MAINT = 177750         162#    1419*   1422    1527    1575    1614    1658    1684    1750    1976    2213    2409    2611
                      3541     5442    5488    5596    5642   10273*  10373*
MANFLG  056012       10318    10338#
MANFL2  056026       10344#
MAPH0 = 170202         398#
MAPH00= 170202         334#     398    1423
MAPH01= 170206         336#     400
MAPH02= 170212         338#     402
MAPH03= 170216         340#     404    2392*   2394*   2594*   2596*   2804*   2819*   2843*
MAPH04= 170222         342#     406
MAPH05= 170226         344#     408
MAPH06= 170232         346#     410
MAPH07= 170236         348#     412
MAPH1 = 170206         400#
MAPH10= 170242         350#
MAPH11= 170246         352#
MAPH12= 170252         354#
MAPH13= 170256         356#
MAPH14= 170262         358#
MAPH15= 170266         360#
MAPH16= 170272         362#
MAPH17= 170276         364#
MAPH2 = 170212         402#
MAPH20= 170302         366#
MAPH21- 170306         368#
MAPH22= 170312         370#
MAPH23= 170316         372#
MAPH24- 170320         374#
MAPH25- 170326         376#
MAPH26- 170332         378#
MAPH27= 170336         380#
MAPH3 - 170216         404#
MAPH30= 170342         382#
MAPH31= 170346         384#
MAPH32= 170352         386#
MAPH33= 170356         388#
MAPH34= 170362         390#
MAPH35= 170366         392#
MAPH36= 170372         394#
MAPH37- 170376         396#
MAPH4 = 170222         406#
MAPH5 = 170226         408#
MAPH6 = 170232         410#
MAPH7 = 170236         412#
MAPL0 = 170200         397#
MAPL00= 170200         333#     397    4547    4707
MAPL01= 170204         335#     399
MAPL02= 170210         337#     401
MAPL03= 170214         339#     403    2391*   2393*   2593*   2595*   2803*   2818*   2842*  11509
MAPL04= 170220         341#     405
MAPL05= 170224         343#     407
MAPL06= 170230         345#     409
MAPL07= 170234         347#     411
MAPL1 = 170204         399#
MAPL10= 170240         349#
MAPL11= 170244         351#
```

```
MAPL12- 170250          353#
MAPL13= 170254          355#
MAPL14= 170260          357#
MAPL15= 170264          359#
MAPL16= 170270          361#
MAPL17= 170274          363#
MAPL2 = 170210          401#
MAPL20= 170300          365#   11368
MAPL21= 170304          367#
MAPL22= 170310          369#
MAPL23= 170314          371#
MAPL24= 170320          373#
MAPL25= 170324          375#
MAPL26= 170330          377#
MAPL27= 170334          379#
MAPL3 = 170214          403#
MAPL30= 170340          381#
MAPL31= 170344          383#
MAPL32= 170350          385#
MAPL33= 170354          387#
MAPL34= 170360          389#
MAPL35= 170364          391#
MAPL36= 170370          393#
MAPL37= 170374          395#
MAPL4 - 170220          405#    1728
MAPL5 = 170224          407#
MAPL6 = 170230          409#
MAPL7 = 170234          411#
MEMERR= 177744          160#    1544*   1548*   1552*   1557*   1593*   1598*   1707*   1781*   2010*   2013    2020*   2245*
                       2247     2254*   2445*   2448    2454*   2644*   2647    2653*   3333    3341    3369*   3614    3641*
                       3900     3908    3924*   4199    4207    4223*   4328    4340    4351*   4463    4475    4486*   4624
                       4636     4647*   4784    4796    4807*   4908    4910*   5334    5370*   5519    5526*   5529    5536*
                       5673     5680*   5683    5690*   5918    9466*  10171   10174   10177*  10234   10241   10277*
MFPT   = 000007         628#    1416
MFPTTR  005216         1415     1463#
MMDES   056256        10025    10427#
MMERR1  035564         6431     6448    6467    6484    6668#
MMERR2  035576         6503     6520    6539    6556    6672#
MMERR3  035610         6670     6675#
MMERR4  035626         6585     6608    6681#
MMERR5  035646         6631     6654    6686#
MMERR6  035666         6684     6690#
MMESRS  066374        10397    11696#
MMPAT1  035542         6312*    6414    6583    6658#  14249   14259
MMPAT2  035544         6313*    6606    6659#  14249
MMPAT3  035546         6314*    6629    6660#  14249   14259
MMPAT4  035550         6315*    6652    6661#  14249   14259
MMRFLG  056006        10304    10336#
MMRFL2  056022        10342#
MMR0   = 177572         183#     187    1624*   1628*   1634*   1641*   1645*   1651*   1687*   1701*   1755*   1768*   1774*
                       1780*    1870*   1969*   2022*   2103*   2206*   2261*   2320*   2402*   2455*   2604*   2660*   2798*
                       2981*    3090*   3229*   3448*   3731*   4029*   4562*   4722*   6781*   9214*   9225*   9231*   9246*
                       9261*    9278*   9286*   9292*  10274*  10588
MMR1   - 177574         184#     188
MMR2     177576         185#     189
MMR3   = 172516         186#     190    1726*   1782*   1869*   1968*   2023*   2102*   2205*   2262*   2319*   2400*   2456*
```

```
                                2602*   2661*   2797*   2980*   3091*   3228*   3447*   3730*   4028*   4561*   -721*   6782*   9213*
                                9293*  10275*
MMSKIP= 104416                  1854    2082    2302    2505    2768    2900    3181    3420    3690    3988    4539    4697   10025#
MMTMP1  035552                  6663#
MMTMP2  035554                  6317    6664#
MMVEC = 000250                   154#   1601*   1620*   1621*   '637*   1680*
MM1     034074                  6312#
MM10    034546                  6430    6432#
MM11    034604                  6440    6445#
MM12    034622                  6447    6449#
MM13    034664                  6459    6464#
MM14    034702                  6466    6468#
MM15    034740                  6476    6481#
MM16    034756                  6483    6485#
MM17    035020                  6495    6500#
MM18    035036                  6502    6504#
MM19    035074                  6512    6517#
MM2     034212                  6331    6333#
MM20    035112                  6519    6521#
MM21    035154                  6531    6536#
MM22    035172                  6538    6540#
MM23    035230                  6548    6553#
MM24    035246                  6555    6557#
MM25    035330                  6573    6583#
MM26    035342                  6584    6588#
MM27    035402                  6596    6606#
MM28    035414                  6607    6611#
MM29    035454                  6619    6629#
MM3     034276                  6347    6355#
MM30    035466                  6630    6634#
MM31    035526                  6642    6652#
MM32    035704                  6653    6656    6696#
MM4     034334                  6361    6370#
MM5     034362                  6376    6384#
MM6     034426                  6390    6399    6400#
MM7     034436                  6402#   6404
MM8     034450                  640?#   6409
MM9     034530                  6423    6428#
MNRNG   067727                 10322   11853#
MONF    056254                  1493*  10387*  10415#
MONTTY  056252                  1496*  10398   10413#
MSER  = 177744                   439#   8848    8849*   8913    8918    8936    8937*   8938    8940    8998    9020*
MSG1    071605                  1469   12053#
MSG2    071644                  1482   12059#
MSG3    071655                  1474   12061#
MSG4    071667                  1476   12063#
MSG5    071720                  1480   12068#
MSIZER  056300                 10026   10443#
MS01    071411                  1386   12031#
MS02    071546                  1387   12047#
MS03    071575                  1393   12051#
MTA101  070316                 11903#
MTA11   066564                 11723#  14222
MTA120  070423                 11916#
MTA124  070704                 11958#
MTA126  070776                 11970#
```

```
MTA131   071056           11981#
MTA134   071221           12002#
MTA135   071255           12008#
MTA17    066631           11731#  11754     14246     14256
MTA20    066665           11740#  14253
MTA21    066674           11743#  14263
MTA43    066761           11756#  14319
MTA45    067034           11765#  14325     14334
MTA5     066502           11713#  14212
MTA50    067112           11777#  14342     14351
MTA77    070140           11880#
MTB120   070453           11922#
MTB126   071024           11975#
MTB131   071140           11991#
MTB135   071305           12014#
MTB17    066636           11733#  14243     14253
MTB21  = 066631           11754#  14265
MTB45    067062           11770#  14325     14328     14334     14337     14342     14345     14351     14354
MTB77    070154           11883#
MTC120   070503           11928#
MTC131   071173           11997#
MTC135   071327           12019#
MTC17    066656           11737#  14243
MTC45    067077           11774#  14328     14334     14337     14345     14351     14354
MTC77    070216           11890#
MTD120   070533           11933#
MTD135   071363           12025#
MTD77    070253           11896#
MTE120   070563           11939#
MTF120   070613           11945#
MTG120   070643           11951#
MO     - 000004             427#   8867
MOM1   = 000014             435#    454#   1880      2112      2328      2527      4270      4403      4564      4724      5971      6078      6082
                           6086     6090      6097      6101      6105      6109      6115      6134      6154      6173      6199      6280      8219
                           8232     8237      8264      8269      8331      8332      8422      8435      8440      8467      8472      8533      8534
                           8577     8578      8586      8587      8619      8620      8707      8708      8770      8771      8807      8988      9176
M1     - 000010             426#   8869
M1MO   = 000014             453#    454      2697      2769      5040
NNDEV    043230            7792     7946#
NNDONE   043476            7833     7990      8005#
NND0     043246            7952#
NND2     043326            7950     7970#
NND3     043412            7972     7988#
NNGRM    043106            7795*    7830*     7908#     7925      7938
NNGRPF   043110            7794*    7809      7826*     7827      7909#
NNGRS    043104            7796*    7829*     7907#     7927      7931
NNRH4    042624            7835#    7992
NNRH4U   042627            7837#    7998*
NNRP4    043010            7816     7882#     7974
NNRP4U   043013            7884#    7980*
NNRS4    042716            7812     7859#     7954
NNRS4U   042721            7861#    7961*
NNSTUP   043120            7798     7920#     7921
NNUD     043102            7799     7812      7816      7905#     7948*     7954*     7974*     7992*
NN1      042420            7792#    7857      7880      7903
NN2      042424            7794#    7962      7981      7999
```

```
NN3      042444       7798#   7831
NN5      042564       7815    7819    7821    7826#
NN6      042620       7828    7833#
NOCNC    056200      10379   10402#
OKSIZ    004750       1385    1398#
PARCNT   056032       5420    5574   10355#
PDMSG1   067140       4884   11782#
PDMSG2   067316       4900   11802#
PIRQ   = 177772         41#   1626*   1629*   1635*   1643*   1646*   1652*
PIRQVE=  000240        153#   1619*   1636*
POWERM   066433      10074   11703#
PP     = 000011       2690#   2754
PPHIAD   012414       2700#   2704
PPLIM    012602       2703*   2706    2708*   2712    2727    2743    2747#
PPLOAD   012412       2699#   2706    2708
PP1      012402       2697#
PP2      012450       2705    2707    2710#
PP3      012470       2716#   2728
PP4      012520       2718    2726#
PP5      012534       2731#   2744
PP6      012570       2735    2742#
PP7      012604       2745    2749#
PR0    = 000000         74#
PR1    - 000040         75#
PR2    = 000100         76#
PR3    = 000140         77#
PR4    = 000200         78#
PR5    = 000240         79#   1656    1673
PR6    = 000300         80#
PR7    = 000340         81#   1620    1682    1702
PS     = 177776         38#     39    9150
PSW    = 177776         39#  10279*  10940*  11100*  11259*  11445*  11572*
PWRVEC=  000024        146#   1306*   1307*  10043*  10044*  10053*  10059*  10071*  10072*
QQERR1   032200       5799    5918#
QQERR2   032256       5926    5929#
QQERR3   032270       5930    5933#
QQERR4   032272       5928    5932    5934#
QQFLG1   031664       5734*   5756    5827#   5907*   5924
QQFLG2   031662       5720*   5825#   5912*
QQGM     031700       5727*   5740    5761    5833#   5905*
QQGS     031676       5726*   5742    5748    5832#   5906*
QQHI     031660       5772*   5781*   5790*   5793*   5814    5819    5822#
QQLO     031656       5771*   5780*   5789*   5792*   5818    5821#
QQPAT1   031666       5716*   5751    5828#   5911*   5929
QQTMP1   031670       5814*   5815*   5816    5829#
QQTMP2   031672       5830#
QQTMP3   031674       5831#
QQ1      031176       5724#   5914
QQ10     031702       5765    5781    5783    5789    5793    5841#
QQ11     031756       5767    5854#
QQ12     031760       5771    5774    5785    5790    5792    5861#
QQ13     032034       5776    5874#
QQ14     032036       5772    5780    5881#
QQ15     032112       5894#
QQ16     032114       5850    5870    5890    5896#   5934
QQ17     032116       5899#
```

```
QQ18     032160           5908    5911#
QQ19     032276           5913    5936#
QQ2      031226           5738#   5909
QQ3      031312           5751#   5760
QQ4      031354           5755    5760#
QQ5      031434           5770    5774#
QQ6      031476           5779    5783#
QQ7      031540           5788    5792#
QQ8      031554           5773    5782    5791    5795#
QQ9      031604           5803    5807#   5903
QQ9.5    031616           5812#
RANDTP   042114           7674*   7677    7706#
RANDWR   041752           7673#  10032
RDCHR  = 104410           9641   10018#
RDLIN  = 104411          10019#
RESMON   056054           1505   10373#  10401
RESREG=  104413           6810    6815    6946    7114    7278    7447    7594    7701    7762    7941   10021#  10208   10616
                         10713   10746   10977   10981   11065   11136   11141   11222   11303   11308   11408   11477   11482
                         11535   11607   11612   11684
RESVEC=  000010            141#   1415*
RHWT     042110           7683*   7687    7697*   7704#
RH4AA    040020           7237#   7369
RH4AA1   040240           7267*   7282#
RH4AA2   040242           7274*   7283#
RH4AA3   040236           7235*   7281#   7285*
RH4AE    004072           1261#  10681*  10682*  11595*
RH4AS    004056           1254#
RH4ASS   036264           6864#
RH4BA    004044           1249#  11594*
RH4BB    040261           7256*   7291#
RH4CC    040262           7262*   7292#
RH4CLR   064352          11632   11683   11687#
RH4CR    036240           6825    6851#   6886    7364*   7376*
RH4CS1   004040           1247#  10675   10681   11604*  11623   11654   11657*  11659   11688
RH4CS2   004050           1251#  10678*  11592*  11622*  11628   11652*  11653*  11656*  11679   11687*
RH4CS3   004074           1262#
RH4CYL   063540          11565#
RH4DA1   063520          11557#  11579*  11596
RH4DA2   063522          11558#  11580*  11646*
RH4DB    004062           1256#
RH4DD    040264           7293#
RH4DFL   057462           7790    7855*  10647*  10718#  10849*
RH4DR    004060           1255#  11596*
RH4DT    004066           1258#  10679
RH4EE    040266           7268*   7294#
RH4ER    004054           1253#  11630   11681
RH4ER1   063500           7372    7845   10641*  11549#  11585*  11631*  11682*
RH4ER2   063502           7377    7849   11550#  11586*  11628*  11679*
RH4ER3   063504           7379    7850   11551#  11587*  11629*  11680*
RH4ER4   063506           7378    7852   11552#  11630*  11681*
RH4FF    040270           7275*   7295#
RH4FLG   063476          10636*  11548#  11567   11603*  11611*  11615*
RH4FT    043114           7790*   7856*   7915#   7989    7997*   8004*
RH4FUN   063514          11555#  11577*  11602
RH4GG    040312           7297    7305#
RH4HAN   063542          10036   11567#
```

```
RH4HH    040325          7257*   7311#
RH4H1    063554         11568   11572#
RH4H2    063670         11592#
RH4H3    064000         11606   11610#
RH4H4    064014         11600   11615#
RH4H5    064030         11610   11616   11621#
RH4H51   064032         11622#  11637   11639
RH4H6    064116         11625   11627   11636#
RH4II    040326          7264*   7312#
RH4JJ    040330          7313#
RH4KK    040332          7269*   7314#
RH4LL    040334          7276*   7315#
RH4MA1   063524         11559#  11581*  11594
RH4MA2   063526         11560#  11582*  11595
RH4MM    040356          7317    7324#
RH4MR1   004064          1257#  11597*
RH4MR2   004046          1250#
RH4NN    040371          7258*   7331#
RH4OO    040372          7265*   7332#
RH4PP    040374          7333#
RH4QQ    040376          7270*   7334#
RH4RB    036276          6870#   7241     7242
RH4RDY   064160         11589   11650#
RH4REG   004036          1246#  10677
RH4REX   004070          1260#  10684
RH4RR    040400          7272*   7335#
RH4SEC   063536         11564#
RH4SS    040464          7355    7361#
RH4ST    004052          1252#  11629    11638    11665    11680
RH4SUN   036252          6857#   7254
RH4S1    064134         11588   11643#
RH4S2    064144         11590   11646#
RH4TMP   063512         11554#  11573*   11576
RH4TRK   063534         11563#
RH4UNI   063516         11556#  11578*   11592    11622    11643*  11651    11653    11656
RH4USE   063510         11553#  11650*   11651*  11652    11687
RH4V     004140          1285#  11599
RH4VEC   063532         11562#  11584*   11617
R'K4WC   004042          1248#  11593*
RH4WCT   063530         11561#  11583*   11593    11647*
RH4XX    040502          7365    7368#
RH4YY    040510          7300    7306     7320     7325     7340     7344     7357     7362     7371#
RH4ZZ    040560          7373    7386#
RH4111   040422          7337    7344#
RH4112   040433          7255*   7349#
RH4113   040434          7263*   7350#
RH4114   040436          7351#
RH4115   040440          7271*   7352#
RH4116   040442          7273*   7353#
RK5AA    040612          7402#   7538
RK5AA1   041052          7436*   7451#
RK5AA2   041054          7443*   7452#
RK5AA3   041050          7400*   7450#    7454*
RK5ASS   036266          6865#
RK5BA    004110          1269#  11290*
RK5BB    041073          7421*   7460#
```

```
RK5CC    041074             7427*   7461#
RK5CLR   062730            11325   11407   11411#
RK5CR    036242             6852#   6878    7533*   7545*
RK5CS1   004104             1267#  11287*  11291*  11298*  11319   11380*  11381   11385*  11387   11412*  11413
RK5CYL   062000            11252#  11350*
RK5DA    004112             1270#  11288*  11292*  11379*  11384*  11411*
RK5DA1   061760            11244#  11266*  11292   11342   11360*
RK5DA2   061762            11245#  11267*
RK5DB    004114             1271#
RK5DD    041076             7431*   7462#
RK5DFL   057463            10648*  10719#  10836*
RK5DS    004100             1265#  10693   11323   11331   11392   11405
RK5EE    041100             7437*   7463#
RK5ER    004102             1266#  11322   11395   11404
RK5ER1   061740             7541   10642*  10829   11236#  11273*  11324*  11406*
RK5ER2   061742             7546   11237#  11274*  11322*  11404*
RK5ER3   061744             7548   11238#  11275*  11323*  11405*
RK5ER4   061746             7547   11239#
RK5FF    041102             7444*   7464#
RK5FLG   061736            10637*  11235#  11254   11297*  11307*  11312*
RK5FT    043115             7916#
RK5FUN   061754            11242#  11264*  11296
RK5GG    041124             7466    7474#
RK5HAN   062002            10037   11254#
RK5HH    041137             7422*   7480#
RK5H1    062014            11255   11259#
RK5H2    062134            11287#
RK5H3    062240            11302   11306#
RK5H4    062254            11294   11311#
RK5H5    062302            11306   11311   11318#
RK5H51   062304            11319#  11330   11332
RK5H6    062346            11321   11329#
RK5II    041140             7429*   7481#
RK5JJ    041142             7432*   7482#
RK5KK    041144             7438*   7483#
RK5LL    041146             7445*   7484#
RK5MA1   061764            11246#  11268*  11290   11365   11376*
RK5MA2   061766            11247#  11269*  11291   11366   11375*
RK5MM    041170             7486    7493#
RK5NN    041203             7423*   7500#
RK5OO    041204             7430*   7501#
RK5PP    041206             7434*   7502#
RK5QQ    041210             7439*   7503#
RK5RB    036300             6871#   7406    7407
RK5RDY   062570            11278   11379#
RK5REG   004076             1264#  10695
RK5RR    041212             7441*   7504#
RK5SEC   061776            11251#  11351*
RK5SS    041276             7524    7530#
RK5SUN   036254             6858#   7419
RK5S1    062364            11277   11336#
RK5S2    062406            11280   11342#
RK5S3    062510            11283   11363#
RK5TMP   061752            11241#  11260*  11263
RK5TRK   061774            11250#  11352*
RK5UNI   061756            11243#  11265*  11288   11336   11339*  11379   11384
```

```
RK5USC   061750         11240#
RK5V     004142          1286#   11293    11313*
RK5VEC   061772         11249#   11271*   11315
RK5WC    004106          1268#   11289*
RK5WCT   061770         11248#   11270*   11289    11363*
RK5XX    041314          7534     7537#
RK5YY    041322          7469     7475     7489     7494     7509     7513     7526     7531     7540#
RK5ZZ    041372          7542     7555#
RK5111   041234          7506     7513#
RK5112   041245          7420*    7518#
RK5113   041246          7428*    7519#
RK5114   041250          7433*    7520#
RK5115   041252          7440*    7521#
RK5116   041254          7442*    7522#
RLWT     042112          7682*    7688     7691     7696*    7705#
RP4AA    037206          7069#    7205
RP4AA1   037446          7103*    7118#
RP4AA2   037450          7110*    7119#
RP4AA3   037444          7067*    7117#    7121*
RP4AS    004000          1230#
RP4ASS   036262          6863#
RP4BA    003766          1225#   10965*
RP4BAE   004032          1243#   10966*
RP4BB    037467          7088*    7127#
RP4CC    037470          7094*    7128#
RP4CCC   004020          1238#
RP4CLR   060764         11000    11064    11068#
RP4CR    036236          6822     6850#    6882     7200*    7212*
RP4CS1   003762          1223#   10665    10974*   10991    11028    11031*   11032    11069
RP4CS2   003772          1227#   10963*   10990*   10996    11026*   11027*   11030*   11060    11068*
RP4CS3   004034          1244#
RP4CYL   060074         10934#
RP4DA    003770          1226#   10967*
RP4DA1   060054         10926#   10947*   10968    11013    11016*
RP4DA2   060056         10927#   10948*   10967    11017*
RP4DB    004004          1232#
RP4DC    004016          1237#   10968*
RP4DD    037472          7098*    7129#
RP4DFL   057461          7789     7901*   10646*   10717#   10807*
RP4DS    003774          1228#   10997    11005    11038    11061
RP4DT    004010          1234#
RP4EC1   004026          1241#
RP4EC2   004030          1242#
RP4EE    037474          7104*    7130#
RP4ER1   060034          7208     7892    10640*   10800    10918#   10953*   10999*   11063*
RP4ER2   060036          7213     7896    10919#   10954*   10996*   11060*
RP4ER3   060040          7215     7897    10920#   10955*   10997*   11061*
RP4ER4   060042          7214     7898    10921#   10998*   11062*
RP4FF    037476          7111*    7131#
RP4FLG   060032         10635*   10917#   10936    10973*   10980*   10984*
RP4FT    043113          7789*    7902*    7914#    7971     7986*
RP4FUN   060050         10924#   10945*   10972
RP4GG    037520          7133     7141#
RP4HAN   060076         10035    10936#
RP4HH    037533          7089*    7147#
RP4H1    060110         10937    10940#
```

```
RP4H2    060230          10963#
RP4H3    060336          10976     10979#
RP4H4    060352          10970     10984#
RP4H5    060366          10979     10985     10989#
RP4H51   060370          10990#    11004     11006
RP4H6    060454          10993     10995     11003#
RP4II    037534           7096*     7148#
RP4JJ    037536           7099*     7149#
RP4KK    037540           7105*     7150#
RP4LA    004002           1231#
RP4LL    037542           7112*     7151#
RP4MA1   060060          10928#    10949*    10965
RP4MA2   060062          10929#    10950*    10966     11021*
RP4MM    037564           7153      7160#
RP4MR    004006           1233#
RP4NN    037577           7090*     7167#
RP4OF    004014           1236#    11055*
RP4OO    037600           7097*     7168#
RP4PP    037602           7101*     7169#
RP4QQ    037604           7106*     7170#
RP4RB    036274           6869#     7073      7074
RP4RDY   060542          10958     11024#
RP4REG   003760           1222#    10667
RP4RR    037606           7108*     7171#
RP4RR1   003776           1229#    10998     11062
RP4RR2   004022           1239#
RP4RR3   004024           1240#
RP4SEC   060072          10933#
RP4SN    004012           1235#
RP4SS    037672           7191      7197#
RP4SUN   036250           6856#     7086
RP4S1    060472          10957     11010#
RP4S2    060502          10959     11013#
RP4S3    060526          10961     11020#
RP4TMP   060046          10923#    10941*    10944
RP4TRK   060070          10932#
RP4UNI   060052          10925#    10946*    10963     10990     11010*    11025     11027     11030
RP4USE   060044          10922#    11024*    11025*    11026     11068
RP4V     004136           1284#    10969
RP4VEC   060066          10931#    10952*    10986
RP4WC    003764           1224#    10964*
RP4WCT   060064          10930#    10951*    10964     11020*
RP4XX    037710           7201      7204#
RP4YY    037716           7136      7142      7156      7161      7176      7180      7193      7198      7207#
RP4ZZ    037766           7209      7222#
RP4111   037630           7173      7180#
RP4112   037641           7087*     7185#
RP4113   037642           7095*     7186#
RP4114   037644           7100*     7187#
RP4115   037646           7107*     7188#
RP4116   037650           7109*     7189#
RR     = 000007           2294#
RRADR1   011444           2332*     2333*     2338      2340      2357      2358      2468*     2469*     2475#
RRADR2   011450           2357*     2358*     2359*     2360*     2361*     2369      2371      2391      2392      2425      2427      2477#    14290
                         14293
RRADR3   011454           2330*     2331*     2338      2340      2359      2361      2457      2459      2461*     2464*     2465*     2470*     2471*
```

```
                         2479#
RRDONE   011460          2377     2380     2482#
RRHIAD   010714          2324#
RRLOAD   010712          2323#    2326*    2369     2371
RR1      010754          2335#    2473
RR10     011406          2458     2460     2464#
RR11     011420          2346     2351     2468#
RR12     011440          2462     2466     2473#
RR2      011012          2349     2355#
RR3      011046          2366#
RR4      011104          2382     2385#
RR5      011230          2416#
RR6      011236          2411     2422#
RR7      011274          2433     2442#
RR8      011314          2436     2438     2448#
RR9      011352          2419     2446     2455#
RSET   = 104414          2049     2277     2482     2677     2749     2867     3159     3373     3653     3949     4246     4383     4518
                         4679     4839     4920     5543     5697     6287     6763     7767     7785     8005    10023#   10259    10375
                        10431
RS4AA    036374          6901#    7037
RS4AA1   036634          6935*    6950#
RS4AA2   036636          6942*    6951#
RS4AA3   036632          6899*    6949#    6953*
RS4AS    003742          1214#
RS4ASS   036260          6794     6861     6862#
RS4BA    003730          1209#   11125*
RS4BAE   003754          1219#   11126*
RS4BB    036655          6920*    6959#
RS4CC    036656          6926*    6960#
RS4CLR   061720         11160    11221    11225#
RS4CR    036234          6792     6819     6849#    6884     7032*    7044*
RS4CS1   003724          1207#   10654    11133*   11151    11194    11197*   11198    11225*   11226
RS4CS2   003734          1211#   11123*   11150*   11156    11192*   11193*   11196*   11217
RS4CS3   003756          1220#
RS4CYL   061044         11094#   11179*
RS4DA    003732          1210#   11127*
RS4DA1   061024         11086#   11107*   11127    11173    11184*
RS4DA2   061026         11087#   11108*
RS4DB    003746          1216#
RS4DD    036660          6930*    6961#
RS4DFL   057460          6791     7788     7878*   10645*   10716#   10779*
RS4DS    003736          1212#   11157    11165    11204    11218
RS4DT    003752          1218#
RS4EE    036662          6936*    6962#
RS4ER    003740          1213#   11158    11219
RS4ER1   061004          7040     7869    10639*   10772    11078#   11113*   11159*   11220*
RS4ER2   061006          7045     7873    11079#   11114*   11156*   11217*
RS4ER3   061010          7047     7874    11080#   11115*   11157*   11218*
RS4ER4   061012          7046     7875    11081#   11158*   11219*
RS4FF    036664          6943*    6963#
RS4FLG   061002         10634*   11077#   11096    11132*   11140*   11144*
RS4FT    043112          7788*    7879*    7913#    7949     7960*    7968*    7979*
RS4FUN   061020         11084#   11105*   11131
RS4GG    036706          6965     6973#
RS4HAN   061046         10034    11096#
RS4HH    036721          6921*    6979#
```

```
RS4H1   061060         11097    11100#
RS4H2   061200         11123#
RS4H3   061300         11135    11139#
RS4H4   061314         11129    11144#
RS4H5   061330         11139    11145    11149#
RS4H51  061332         11150#   11164    11166
RS4H6   061416         11153    11155    11163#
RS4II   036722          6928*    6980#
RS4JJ   036724          6931*    6981#
RS4KK   036726          6937*    6982#
RS4LA   003744          1215#
RS4LL   036730          6944*    6983#
RS4MA1  061030         11088#   11109*   11125
RS4MA2  061032         11089#   11110*   11126    11188*
RS4MM   036752          6985     6992#
RS4MR   003750          1217#
RS4NN   036765          6922*    6999#
RS4OO   036766          6929*    7000#
RS4PP   036770          6933*    7001#
RS4QQ   036772          6938*    7002#
RS4RB   036272          6868#    6905     6906     7709     7752
RS4RDY  061526         11118    11190#
RS4REG  003722          1206#   10656
RS4RR   036774          6940*    7003#
RS4SEC  061042         11093#
RS4SS   037060          7023     7029#
RS4SUN  036246          6793     6855#    6918
RS4S1   061434         11117    11170#
RS4S2   061444         11120    11173#
RS4S3   061512         11121    11187#
RS4TMP  061016         11083#   11101*   11104
RS4TRK  061040         11092#   11178*
RS4UNI  061022         11085#   11106*   11123    11150    11170*   11191    11193    11196
RS4USE  061014         11082#   11190*   11191*   11192    11225
RS4V    004134          1283#   11128
RS4VEC  061036         11091#   11112*   11146
RS4WC   003726          1208#   11124*
RS4WCT  061034         11090#   11111*   11124    11187*
RS4XX   037076          7033     7036#
RS4YY   037104          6968     6974     6988     6993     7008     7012     7025     7030     7039#
RS4ZZ   037154          7041     7054#
RS4111  037016          7005     7012#
RS4112  037027          6919*    7017#
RS4113  037030          6927*    7018#
RS4114  037032          6932*    7019#
RS4115  037034          6939*    7020#
RS4116= 037036          6941*    7021#
SAVREG= 104412          6803     6903     7071     7239     7404     7568     7676     7742     7920    10020#   10193    10582    10644
                       10739    10943    11103    11262    11448    11575

SDPAR0= 172260           271#
SDPAR1= 172262           272#
SDPAR2= 172264           273#
SDPAR3= 172266           274#
SDPAR4= 172270           275#
SDPAR5= 172272           276#
SDPAR6= 172274           277#
```

```
SDPAR7= 172276          278#
SDPDR0= 172220          249#
SDPDR1= 172222          250#
SDPDR2= 172224          251#
SDPDR3= 172226          252#
SDPDR4= 172230          253#
SDPDR5= 172232          254#
SDPDR6= 172234          255#
SDPDR7= 172236          256#     9106
SETBLE= 036260         6861#
SETMP   057536        10737*    10740     10749#
SETREG  057500        10655     10666     10676     10683     10694     10704     10737#
SIPAR0= 172240          260#
SIPAR1= 172242          261#
SIPAR2= 172244          262#
SIPAR3= 172246          263#
SIPAR4= 172250          264#
SIPAR5= 172252          265#
SIPAR6= 172254          266#
SIPAR7= 172256          267#
SIPDR0= 172200          238#     9083
SIPDR1= 172202          239#
SIPDR2= 172204          240#
SIPDR3= 172206          241#
SIPDR4= 172210          242#
SIPDR5= 172212          243#
SIPDR6= 172214          244#
SIPDR7= 172216          245#
SIZDEV  057066         6786      7787     10634#
SIZE  = 104417         1871      2104      2322      2522      2698      2770      2922      3183      3437      3692      3990      7714     10026#
SIZEHI= 177762          170#
SIZELO= 177760          168#     1383      1389
SIZRH4  060022        10686     10849#
SIZRK5  057710        10696     10812#
SIZRP4  057624        10668     10785#
SIZRS4  057540        10657     10756#
SIZTM1  057466        10651*    10712     10723#
SIZTM2  057470        10724#
SIZTM3  057472        10725#
SIZTM4  057474        10726#
SIZTM5  057476        10727#
SIZUBE  057774        10706     10841#
SKAD    055572         1846*     2077*     2296*     2501*     2692*     2764*     2895*     3177*     3416*     3685*     3981*     4265*     4398*
                       4536*     4693*     4868*     4875      4933*     5117*     5234*     5412*     5566*     5710*     5964*     6306*     6761*
                       7783*    10260     10262#    10412     10432
SKBADR  055664        10027     10295#
SKBCNR  055726        10029     10311#
SKBERR  055710        10028     10304#
SKBHMR  055762        10031     10325#
SKBMNR  055744        10030     10318#
SKIPT = 104415         5438      5484      5592      5638     10024#    10230     10248     10332
SKPBAD= 104420        10027#
SKPBCN= 104422        10029#
SKPBER= 104421        10028#
SKPBHM= 104424        10031#
SKPBMN= 104423        10030#
```

```
SKRNG    055776        10302   10309   10316   10323   10331#
SPUR     055440         1510    1601    1706    1849    2080    2299    2504    2695    2767    2898    3191    3327    3329
                        3335    3419    3513    3616    3617    3635    3637    3687    3902    3983    4201    4330    4333
                        4465    4468    4626    4629    4786    4789    4871    4914    4935    5119    5236    5243    5341
                        5345    5415    5521    5531    5569    5675    5685    5713    5969    6308   10233#  10240   10269
SR0    = 177572          187#  10124   10139*  10147*
SR1    = 177574          188#
SR2    = 177576          189#
SR3    = 172516          190#  10136*
SS     = 000010         2499#
SSADR1   012326         2529*   2530*   2535*   2536*   2537*   2545    2546    2666#
SSADR2   012332         2545*   2546*   2547*   2548*   2551    2553    2570    2572    2593    2594    2624    2626    2668#
                       14296   14299
SSCNST   012342         2551    2553    2674#
SSHIAD   011604         2524#
SSLOAD   011602         2523#   2525*   2570    2572
SSMASK   012336         2539*   2540*   2547    2548    2655*   2656*   2671#
SS1      011622         2529#
SS10     012264         2564    2581    2619    2645    2655#
SS11     012346         2663    2677#
SS2      011642         2534#   2664
SS3      011662         2532    2539#
SS4      011704         2543    2545#   2659
SS5      011772         2559    2562    2567#
SS6      012030         2578    2583    2587#
SS7      012152         2621#
SS8      012210         2613    2632    2641#
SS9      012226         2635    2637    2647#
STACK  = 001400           32#     33      34      35     458#   1298    1528    1576    1623    1640    1675    1686    1727
                        1735    1754    4890    4921   10272
START    004146          469    1291#  10076
STKLMT=  177774           40#
STOP     056144        10389   10393#
SUPSTK=  000700           34#
SWR      001540          536#   1296    1318*   1320    1326*   1333*   1347    1738    4873    9362    9376    9378    9386
                        9393    9432    9439    9451    9455    9538    9575*  10051   10064*  10402   10427
SWREG    000176          467#   1326    1347    9538    9551   10402
SW0    = 000001          109#
SW00   = 000001           99#    109
SW01   = 000002           98#    108
SW02   = 000004           97#    107
SW03   = 000010           96#    106
SW04   = 000020           95#    105
SW05   = 000040           94#    104
SW06   = 000100           93#    103
SW07   = 000200           92#    102
SW08   = 000400           91#    101
SW09   = 001000           90#    100
SW1    = 000002          108#
SW10   = 002000           89#
SW11   = 004000           88#
SW12   = 010000           87#   4873
SW13   = 020000           86#
SW14   = 040000           85#   1738
SW15   = 100000           84#
SW2    = 000004          107#
```

```
SW3    = 000010            106#
SW4    = 000020            105#
SW5    = 000040            104#
SW6    = 000100            103#
SW7    = 000200            102#   10427
SW8    = 000400            101#
SW9    = 001000            100#
SYSTID= 177764             172#
S0     = 000020            425#    8222    8426    8865    8952    9025    9179
S0M0M1= 000034             434#     452#   3188    3292    3697    3879    3995    4178    4962    5124    5211    6006    6334
                          8205     8317    8376    8408    8517    8630    8691    8779    8802    8851    8956    8976    9029
                          9170
S0M1   = 000030            450#    2932    3096    3450    5241    5315    5424    5578    5727    5906    7796    7830
S1     = 000040            424#    8223    8425    8559    8605    8626    8727    8753    8775    8908    9183
S1M0   = 000044            449#    2933    3095    3584    5240    5314    5470    5624    5726    5905    7795    7829
S1M0M1= 000054             433#     451#   3189    3291    3698    3878    3996    4179    4967    5123    5212    6009    6337
                          8206     8318    8375    8409    8518    8629    8692    8778    8804    8852    8955    8977    9028
                          9172
TAB    = 000011            448#   11711   11716   11723   11737   11740   11746   11756   11765   11774   11777   11903   11908
                         11970    11975   11981   11991   11997   12002   13577   13586   13594   13603   13615   13624   13636
                         13645    13651   13657   13662   13667   13673   13679   13685   13694   13702   13708   13717   13725
                         13745    13757   13765   13770   13777   13787   13797   13803   13811   13826   13886   13896   13917
                         13930    13938   13949   13955   13962   13969   13977   13982
TBITVE= 000014             142#
TFSTR1= 140000             455#    4941    5137    5155    5163    5186    5254    5274    5796    5987    6320    6411    6562
                          7802     7840    7864    7887    7934
TESTR2= 142000             456#    4943    5747    5795    5807    5989    6322    6413    6564    7923
TESTR3= 144000             457#    4945
TKVEC  = 000060            149#    1496    1505*   1506*  10398*  10401*
TLOC   = 116310          14415#   14416#  14417#  14418
TMP      045332           8345*    8357    8360    8379#
TOP      005342           1494     1503#
TPVFC  = 000064            150#
TRAPVE= 000034             148#    1304*   1305*
TRTVEC= 000014             143#
TSTDAT   116310           8216     8238    8273    8333    8342    8419    8441    8476    8536    8548    8589    8598    8710
                          8719     8745   14419#
TSTDT1   064366          11690#
TST1     005412           1518#
TST10    011462           2296     2497#
TST11    012350           2501     2688#
TST12    012606           2692     2760#
TST13    013246           2764     2892#
TST14    014420           2895     3172#
TST15    015450           3177     3412#
TST16    016540           3416     3680#
TST17    020162           3685     3976#
TST2     005610           1569#
TST20    021610           3981     4261#
TST21    022374           4265     4394#
TST22    023172           4398     4532#
TST23    024060           4536     4689#
TST24    024760           4693     4865#
TST25    025214           4868     4930#
TST26    026322           4933     5114#
TST27    027032           5117     5231#
```

```
TST3     005744          1610#
TST30    030012          5234    5408#
TST31    030464          5412    5562#
TST32    031140          5566    5708#
TST33    032276          5710    5961#
TST34    034042          5964    6303#
TST35    035704          6306    6759#
TST36    042352          6761    7781#
TST37    043520          8023#
TST4     006444          1724#
TST40    043724          8068#
TST41    044040          8086    8098#
TST42    044120          8124#
TST43    044232          8153#
TST44    044370          8200#
TST45    045014          8295    8314#
TST46    045334          8374    8403#
TST47    045764          8498    8514#
TST5     006742          1842#
TST50    046442          8627    8644#
TST51    046530          8689#
TST52    047074          8776    8798#
TST53    047244          8826    8835    8847#
TST54    047646          8953    8973#
TST55    050102          9041#
TST56    050236          9046    9077#
TST57    050372          9082    9112#
TST6     007650          1846    2073#
TST60    050526          9117    9147#
TST7     010550          2077    2292#
TT     = 000012          2762#
TTHIAD   012654          2772#   2775
TTLIM    013242          2774*   2777    2779*   2811    2835    2861    2865#
TTLOAD   012652          2771#   2777    2779
TT1      012642          2769#
TT2      012710          2776    2778    2780#
TT3      013054          2816#   2836
TT4      013126          2826    2833#
TT5      013146          2840#   2862
TT6      013224          2852    2859#
TT7      013244          2863    2867#
TVADFL   056722         10528*  10540*  10568#  10586
TVADHI   056726         10575#  10585*  10608*
TVADLO   056724         10574#  10584*  10609*  10610
TYPDS  = 104405          9325   10013#  10505
TYPE   = 104401          1341    1360    1386    1387    1388    1393    1469    1474    1476    1480    1482    1740    4883
                         4899    9323    9326    9434    9442    9549    9550    9553    9566    9577    9596    9645    9648
                         9652    9758    9855    9931   10009#  10073   10300   10307   10314   10321   10328   10380   10396
                        10408   10467   10482   10484   10488   10490   10514   10534   10550   10556   10614
TYPOC  = 104402          9552   10010#  10474   10500
TYPON  = 104404         10012#
TYPOS  = 104403          1390    1395   10011#  10521
TYPVAD   056730         10529   10541   10582#
T.END    005144          1445    1448#
T1       00477(          1419#   1464
T2       00504(          1426    1429#
T3       005072          1431     1434     1437#
T4       005124          1440     1443#
UBCLR    063462         11497    11534    11538#
UBEAA    041422          7568#    7640
UBEAA1   041556          7575*    7598#
UBEAA2   041560          7578*    7599#
UBEAA3   041554          7566*    7597#    7601*
UBEASS   036270          6866#
UBEBA    004124          1277#   11466*
UBEBB    041576          7606#
UBECC    004122          1276#   11465*
UBECCC   041600          7592*    7607#
UBECLR   004130          1279#   11522*   11538*
UBECR    036244          6853#    6880     7636*    7646*
UBECR1   004126          1278#   10841*   10842    11474*   11490    11494    11524    11531    11539
UBECR2   004132          1280#   11467*   11495    11532
UBECYL   063014         11438#
UBEDA1   062774         11430#   11452*
UBEDA2   062776         11431#   11453*   11468
UBEDB    004120          1275#   10703    11468*
UBEDD    041602          7576*    7608#
UBEDFL   057464         10649*   10720#   10844*
UBEEE    041604          7579*    7609#
UBEER1   062754          7642    10643*   11422#   11458*   11496*
UBEER2   062756          7647    11423#   11459*   11494*   11531*
UBEER3   062760          7648    11424#   11460*   11495*   11532*   11533*
UBEER4   062762         11425#
UBEFF    041624          7612     7618#
UBEFLG   062752         10638*   11421#   11440    11473*   11481*   11485*
UBEFT    043116          7917#
UBEFUN   062770         11428#   11451*   11472
UBEGG    041636          7623#
UBEHAN   063016         10038    11440#
UBEHH    041640          7591*    7624#
UBEH1    063030         11441    11445#
UBEH2    063134         11465#
UBEH3    063226         11476    11480#
UBEH4    063242         11470    11485#
UBEH5    063256         11480    11486    11489#
UBEH51   063260         11490#   11502
UBEH6    063322         11492    11501#
UBEII    041642          7577*    7625#
UBEJJ    041644          7580*    7626#
UBEKK    041664          7628     7634#
UBELL    041700          7637     7640#
UBEMA1   063000         11432#   11454*   11506    11517*
UBEMA2   063002         11433#   11455*   11467    11507    11516*
UBERB    036302          6872#    7570     7572
UBERDY   063406         11461    11522#
UBEREG   004116          1274#   10705
UBESEC   063012         11437#
UBESUN   036256          6859#
UBES1    063332         11462    11506#
UBETMP   062766         11427#   11446*   11449
UBETRK   063010         11436#
UBEUNI   062772         11429#
```

```
UBEUSE  062764        11426#
UBEV    004144         1287#   11469
UBEVEC  063006        11435#   11457*   11487
UBEWCT  063004        11434#   11456*   11465    11518*   11519*
UBEYY   041704         7614     7618     7630     7634     7642#
UBEZZ   041744         7643     7653#
UCB    = 001000         422#    8717     8744
UDPAR0= 177660          227#
UDPAR1= 177662          228#
UDPAR2= 177664          229#
UDPAR3= 177666          230#
UDPAR4= 177670          231#
UDPAR5= 177672          232#
UDPAR6= 177674          233#
UDPAR7= 177676          234#
UDPDR0= 177620          205#
UDPDR1= 177622          206#
UDPDR2= 177624          207#
UDPDR3= 177626          208#
UDPDR4= 177630          209#
UDPDR5= 177632          210#
UDPDR6= 177634          211#
UDPDR7= 177636          212#    9141
UIPAR0= 177640          216#
UIPAR1= 177642          217#
UIPAR2= 177644          218#
UIPAR3= 177646          219#
UIPAR4= 177650          220#
UIPAR5= 177652          221#
UIPAR6- 177654          222#
UIPAR7= 177656          223#
UIPDR0= 177600          194#    9118
UIPDR1= 177602          195#
UIPDR2= 177604          196#
UIPDR3= 177606          197#
UIPDR4= 177610          198#
UIPDR5= 177612          199#
UIPDR6= 177614          200#
UIPDR7= 177616          201#
USESTK= 000600           35#
UU     = 000016         3682#    3954
UUADR1  017652          3735*    3736*    3742     3743     3771*    3772*    3795*    3796*    3809*    3810*    3815     3816     3844*
                        3845*    3868*    3869*    3889#
UUADR2  017656          3733*    3734*    3744     3746     3797*    3807*    3808*    3817     3819     3870*    3891#
UUADR3  017662          3742*    3743*    3744*    3745*    3746*    3754     3756     3775     3792     3793     3815*    3816*    3817*
                        3818*    3819*    3827     3829     3848     3864     3865     3893#    3909     3910
UUDONE  020160          3876     3941     3946     3949#
UUERR1  017674          3738     3900#    3927
UUERR2  017710          3901     3904#
UUERR3  020030          3917     3923#
UUERR4  020032          3922     3924#
UUFLG1  017644          3696*    3791     3863     3875     3877*    3882#    3907
UUFLG2  017646          3700*    3739*    3801*    3812*    3874*    3885#    3929     3932     3935     3938     3945
UUFLG3  017650          3689*    3887#    3940     3942*
UUGM    017670          3698*    3706     3803     3879*    3897#
UUGS    017666          3697*    3704     3710     3878*    3896#
```

```
UUHIAD   016604        3694#
UULOAD   016602        3693#   3754    3756    3827    3829
UUTMP    017672        3898#   3916*   3926
UU0      016550        3683#   3701
UU1      016626        3700#   3880    3947
UU10     017406        3824#
UU11     017444        3838    3844#
UU12     017460        3835    3840    3848#
UU13     017552        3861    3868#   3937
UU14     017610        3875#   3943
UU2      016642        3704#   3708
UU3      017042        3740#   3773    3800
UU4      017076        3751#
UU5      017134        3765    3771#
UU6      017150        3762    3767    3775#
UU7      017244        3790    3795#   3931
UU8      017316        3807#   3934
UU9      017352        3813#   3846    3873
VCIP   = 010000         419#   8048    8059    8076    8102    8107    8126    8134    8142    8155    8165    8168    8177
                       8203    8258    8262    8320    8406    8461    8465    8522    8646    8696    8856    8981    9165
VPBP     050566        9157    9159    9161#   9168
VPBPA    050602        9164#   9300
VPBPE    051360        9160    9299    9301#
VSCM     047650        8974#
VSCMA    047662        8978#   9024    9031
VSG0     044372        8201#   8209    8233    8265
VSG0A    044374        8202#   8296
VSG1     045336        8404#   8412    8436    8468
VSG1A    045340        8405#   8499
VSIU   = 020000         418#   8041    8042    8054    8055    8071    8078    8079    8084    8085    8129    8137    8159
                       8167    8171    8227    8260    8293    8367    8430    8463    8496    8622    8772    8930    8948
                       9022
VSPE   = 100000         416#   8900    9004    9013    9014
VV     = 000017        3658    3978#
VVADR1   021300        4033*   4034*   4041    4042    4071*   4072*   4096*   4097*   4110*   4111*   4116    4117    4145*
                       4146*   4169*   4170*   4187#
VVADR2   021304        4031*   4032*   4043    4045    4098*   4108*   4109*   4118    4120    4171*   4189#
VVADR3   021310        4041*   4042*   4043*   4044*   4045*   4053    4055    4075    4092    4093    4116*   4117*   4118*
                       4119*   4120*   4128    4130    4149    4165    4166    4191#   4208    4209
VVDONE   021606        4177    4239    4243    4246#
VVERR1   021322        4036    4199#   4226
VVERR2   021336        4200    4203#
VVERR3   021456        4216    4222#
VVERR4   021460        4221    4223#
VVFLG1   021272        3994*   4091    4164    4176    4180*   4183#   4206
VVFLG2   021274        3998*   4037*   4102*   4113*   4175*   4184#   4227    4230    4233    4236    4242
VVFLG3   021276        3986*   4185#   4238    4240*
VVGM     021316        3996*   4004    4104    4178*   4195#
VVGS     021314        3995*   4002    4008    4179*   4194#
VVHIAD   020226        3992#
VVLOAD   020224        3991#   4053    4055    4128    4130
VVTMP    021320        4197#   4215*   4225
VV0      020172        3979#   3999
VV1      020250        3998#   4181    4244
VV10     021034        4125#
VV11     021072        4139    4145#
```

```
VV12     021106          4136    4141    4149#
VV13     021200          4162    4169#   4235
VV14     021236          4176#   4241
VV2      020264          4002#   4006
VV3      020466          4039#   4073    4101
VV4      020522          4050#
VV5      020560          4064    4071#
VV6      020574          4061    4066    4075#
VV7      020670          4089    4096#   4229
VV8      020742          4108#   4232
VV9      021000          4114#   4147    4174
WAITLP   036304          6789    6877#   6879    6881    6883    6885    6887
WRRAND=  104425          6948    7116    7280    7449    7596    10032#
X      = 000005          1844#   2281
XADR1    007626          1884*   1885*   1890    1892    1910    1911    2035*   2036*   2041#
XADR2    007632          1910*   1911*   1912*   1913*   1914*   1922    1924    1957    1958    1991    1993    2043#   14272
                        14275
XADR3    007636          1882*   1883*   1890    1892    1912    1914    2024    2026    2028*   2031*   2032*   2037*   2038*
                         2045#
XADR4    007642          2047#
XDONE    007646          1930    1933    2049#
XHIADR   007100          1876#
XLOADR   007076          1875#   1877*   1922    1924
XX     = 000006          2075#   2486
XXADR1   010526          2115*   2116*   2121*   2122*   2123*   2131    2132    2267#
XXADR2   010532          2131*   2132*   2133*   2134*   2138    2140    2158    2160    2194    2195    2226    2228    2269#
                        14278   14281
XXCNST   010542          2138    2140    2274#
XXHIA    010006          2106#
XXLOA    010004          2105#   2107*   2158    2160
XXMASK   010536          2125*   2126*   2133    2134    2256*   2257*   2271#
XX1      010024          2115#
XX10     010464          2151    2169    2221    2246    2256#
XX11     010546          2264    2277#
XX2      010044          2120#   2265
XX3      010064          2118    2125#
XX4      010106          2129    2131#   2260
XX5      010174          2146    2149    2155#
XX6      010232          2166    2171    2175#
XX7      010354          2215    2223#
XX8      010412          2234    2243#
XX9      010426          2237    2239    2247#
X1       007140          1887#   2039
X10      007570          2025    2027    2031#
X11      007602          1898    1903    2035#
X12      007622          2029    2033    2039#
X2       007176          1901    1908#
X3       007232          1919#
X4       007270          1935    1938#
X5       007414          1982#
X6       007422          1978    1988#
X7       007460          1999    2008#
X8       007476          2002    2004    2013#
X9       007534          1986    2011    2022#
ZADHI    013370          2924#
ZADLO    013366          2923#   2992    2994    3020    3022
```

```
ZCMTBL   014310          29_1      2959      2967      3129#
ZFLG1    014164          2926*     3049      3069      3092      3094*     3100#
ZFLG2    014166          2935*     2944      2954      2962      3085*     3097*     3102#
ZGM      014170          2933*     2973      3096*     3108#
ZGS      014172          2932*     2975      3095*     3109#
ZTABLE   014206          2949*     2950*     2957*     2958*     2965*     2966*     2983      3006      3119#     3129      3143      3145      3150
ZTABOT   014306          3078      3082      3120#
ZTHR     014176          2972      3111#     3151
ZTMP1    014202          3114#     3134*     3141      3142
ZTMP2    014204          3115#     3135*
Z1       013416          2944#     3086      3098
Z10      013674          3025      3028      3034#
Z11      013762          3048      3054#
Z12      014062          3013      3068      3076#
Z13      014074          3030      3081#
Z14      014116          2970      3002      3090#
Z15      014416          3093      3159#
Z2       013444          2945      2954#
Z3       013474          2955      2962#
Z4       013524          2963      2970#
Z5       013530          2952      2960      2968      2972#
Z7       013574          2987#     3083
Z8       013634          3012#     3079
Z9       013640          3015#
$APTHD   001400           492       498#
$ASTAT=  ****** U        9505      9520
$ATYC    052244          9476      9478#
$ATY1    052220          9474#
$ATY3    052226          9475#     9743
$ATY4    052236          9447      9477#
$AUTOB   001534           533#     1351*     9546      9667      10406
$BDADR   001522           528#
$BDDAT   001526           530#
$BELL    001706           590#     9434      9469
$CHARC   053614          9760*     9770*     9777      9786*     9791#
$CKSWR   052466          9538#    10017
$CMTAG   001500           516#     1293      1294      1302      1308      1309      1310
$CM1   = 000024           548#      549#      550#      551#      552#      553#      554#      555#      556#      557#      558#      559#      560#
                          561#      562#      563#      564#      565#      566#      567#      568#
$CM2   = 000050           548#      549#      550#      551#      552#      553#      554#      555#      556#      557#      558#      559#      560#
                          561#      562#      563#      564#      565#      566#      567#      568#
$CM3   = 000024           546#      548
$CM4   = 000024           568#      569#      570#      571#      572#      573#      574#      575#      576#      577#      578#      579#      580#
                          581#      582#      583#      584#      585#      586#      587#      588#
$CNTLG   053213          9549      9662#    10408
$CNTLU   053206          9566      9661#
$CORE    055130         10122     10150#
$CPUOP   001744           613#
$CRLF    001713           592#     1388      9442      9469      9577      9661      9759      9794     10468     10485     10491     14243     14249
                        14253     14259     14265
$CROUT   055160         10150     10157#
$DBLK    054262          9897      9931      9939#
$DB20    055272         10193#    10511     10548     10611
$DFVCT   001726           604#
$DUAGN   051474          9319      9328      9332      9338#
$DTBL    054252          9900      9935#
```

```
$ENDAD  051464           477    1339    1358    9334#   9462
$ENDCT  051414          1308    9321#
$ENDMG  051503          9323    9342#
$ENULL  051500          9326    9341#
$ENV    001736           609#   1345    9444    9483    9507    9738
$ENVM   001737           610#   1331    9485    9740    9745
$EOP    051360          9311#
$EOPCT  051406          1308*   9318#   9322
$EOT    006742          1737    1739    1741    1783#
$ERFLG  001503           519#   9351    9382    9384    9390*   9412    9429*   9469
$ERMAX  001515           525#   1311*   9384    9407*   9412
$ERROR  052006          1302    9427#
$ERRPC  001516           526#   9436*   9437*   9438    9469   10473   14201   14212   14227   14234   14239   14272   14275
                       14278   14281   14284   14287   14290   14293   14296   14299   14302   14307   14314   14319   14325
                       14342   14359   14361   14362   14364   14367   14370   14372   14376   14381   14413
$ERRTB  001752           643#  10479
$ERTTL  001512           523#   9435*   9469
$ESCAP  001704           589#   1310*   1519*   1570*   9406*   9458    9460    9469
$ETABL  001736           608#
$ETEND  001746           504     620#
$FATAL  001720           601#   9511*
$FFLG   052464          9474*   9477*   9505    9514*   9522#
$FILLC  001556           544#   9763    9794
$FILLS  001555           543#   9794
$GDADR  001520           527#
$GDDAT  001524           529#
$GET42  051436          9327#  10395
$GTSWR  052536          9550#  10015
$HD   = 000003            10      11
$HIBTS  001400           499#
$HINUM  054370          6911*   6914    7079*   7082    7247*   7250    7412*   7415    7583*   7586    7684*   7695    7744*
                        7747    9957    9965    9970*   9975#

$ICNT   001504           520#   9397*   9398    9400*   9411
$ILLUP  054662         10043   10059   10078#
$INTAG  001535           534#   9578    9667
$ITEMB  001514           524#   3284*   5307*   9438*   9446    9469   10256*  10471
$KTNEX  055122         10123   10149#
$KTOUT  055112         10140   10146#  10178
$KT11   054752          1379*  10121#  10125*  10149*
$LF     001714           593#   9469    9652    9661    9794
$LFLG   052463          9515*   9521#
$LONUM  054372          6912*   6915    7080*   7083    7248*   7251    7413*   7416    7584*   7587    7685*   7743*   7746
                        9956    9963    9969*   9976#

$LPADR  001506           521#   1312*   9388*   9404*   9409    9411
$LPERR  001510           522#   1313*   1521*   1571*   1622*   1638*   1653*   1679*   1751*   1938*   2129*   2385*   2543*
                        4958*   4990*   5006*   5022*   5038*   5046*   5139*   5157*   5165*   5188*   5242*   5256*   5282*
                        5419*   5469*   5573*   5623*   5724*   5803*   6004*   6331*   6399*   9388    9405*   9411    9457
$LSTAD  055266         10164*  10179#
$LSTBK  055270          1381*   1383    1394   10165*  10180#  10446
$MAIL   001716           500     504     599#   1330    1345    9403    9444    9738
$MBADR  001402           500#
$MFLG   052462          9475*   9481    9516*   9520#
$MNEW   053231          9553    9665#
$MSGAD  001732           606#   9491*   9494
$MSGLG  001734           607#   9496*
$MSGTY  001716           600#   9489    9497*   9509    9513*
```

```
$MSWR    053220          9550    9663#
$MTMOU   055234         10141   10171#
$MXCNT   052004          9401    9411#
$NULL    001554           542#   9765    9794
$NWTST=  000001          1512#   1514    1560#   1562    1603#   1605    1714#   1716    1784#   1786    2051#   2053    2279#
                         2281    2484#   2486    2679#   2681    2752#   2754    2870#   2872    3162#   3164    3375#   3377
                         3656#   3658    3952#   3954    4249#   4251    4386#   4388    4520#   4522    4681#   4683    4842#
                         4844    4923#   4925    5107#   5109    5224#   5226    5391#   5393    5545#   5547    5700#   5702
                         5939#   5941    6289#   6291    6701#   6703    7772#   7774    8016#   8018    8061#   8063    8091#
                         8093    8116#   8118    8146#   8148    8181#   8183    8299#   8301    8382#   8384    8502#   8504
                         8635#   8637    8669#   8671    8784#   8786    8841#   8843    8959#   8961    9037#   9039    9073#
                         9075    9108#   9110    9143#   9145
$OCNT    054042          9827*   9856*   9869#
$OCTVL   055374         10195   10220#
$OMODE   054044          9822*   9826*   9831    9834*   9845*   9871#
$OVER    051770          9363    9381    9389    9399    9408#
$PASS    001724           603#   1330*   1736    9315*   9316*   9324    9341    9395    9412
$PASTM   001406           502#
$PWRAD   054656         10076#
$PWRDN   054516          1306   10043#  10071
$PWRMG   054652         10074#
$PWRUP   054570         10053   10059#
$QUES    001712           591#   9469    9596    9645    9661    9794
$RAND    054272          6913    7081    7249    7414    7585    7694    7745    9952#
$RDCHR   052750          9609#  10018
$RDDEC=  ****** U       10020
$RDLIN   053070          9637#  10019
$RDOCT=  ****** U       10020
$RDSZ =  000010          9630#
$REGAD   001560           546#
$REG0    001562           548#   8027*   8033*   9038*   8044*   8051*   8057*   8073*   8081*   8087*   8104*   8110*   8131*
                         8139*   8'61*   8173*   8229*   8243*   8279*   8348*   8359*   8432*   8446*   8482*   8557*   8603*
                         8658*   8725*   8751*   8813*   8828*   8880*   8895*   8902*   8917*   8932*   8940*   8997*   9007*
                         9016*   9051*   9058*   9065*   9087*   9094*   9101*   9122*   9129*   9136*   9220*   9241*   9273*
                        14361   14362   14364   14367   14370   14372
$REG1    001564           549#   8244*   8280*   8349*   8352*   8360*   8447*   8483*   8558*   8561*   8604*   8607*   8726*
                         8729*   8752*   8755*   8814*   8829*   8918*   8998*   9052*   9059*   9066*   9088*   9095*   9102*
                         9123*   9130*   9137*   9221*   9242*   9274*  14362   14364   14367   14370   14372
$REG10   001602           556#
$REG11   001604           557#
$REG12   001606           558#
$REG13   001610           559#
$REG14   001612           560#
$REG15   001614           561#
$REG16   001616           562#
$REG17   001620           563#
$REG2    001566           550#   8245*   8281*   8353*   8361*   8448*   8484*   8562*   8608*   8730*   8756*   8815*   8830*
                         8919*   9222*   9243*   9275*  14362   14364   14367   14372
$REG20   001622           564#
$REG21   001624           565#
$REG22   001626           566#
$REG23   001630           567#
$REG3    001570           551#   8920*   9223*   9244*   9276*  14364   14367   14372
$REG4    001572           552#   9224*   9245*   9277*  14364   14372
$REG5    001574           553#
$REG6    001576           554#
```

```
$REG7    001600              555#
$RESRE   053300             9702#   10021
$RTNAD   051476             9340#
$R2A   = ****** U          10022
$SAVRE   053242             9686#   10020
$SAVR6   054666            10052*   10060  10061*  10062*  10080#
$SCOPE   051520             1300    9360#
$SETUP=  000137              442#    1299    1300    1302    1304    1306    1308    1309    1310    1312    1339    1342    9313
                            9361     9428    9454    9462    9533    9667
$SIZE    054670             1380    10105#
$SIZEX   055164            10148    10158#
$STUP  = 177777              442#
$SVLAD   051734             9371     9402#
$SVPC  = 000204              475#     480
$SWR   = 167400               10       11#     12#     19      20      21      22      23      24      25      26     588     589
                             590     1309    1310    1312    1313    1519    1570    1611    1725    1843    2074    2293    2498
                            2689     2761    2893    3173    3413    3681    3977    4262    4395    4533    4690    4866    4931
                            5115     5232    5409    5563    5709    5962    6304    6760    7782    8024    8069    8099    8125
                            8154     8201    8315    8404    8515    8645    8690    8799    8848    8974    9042    9078    9113
                            9148     9308    9314    9329    9339    9341    9352    9353    9354    9355    9356    9362    9374
                            9376     9377    9382    9383    9384    9391    9392    9393    9405    9408    9411    9419    9420
                            9421     9422    9423    9432    9439    9451    9455    9469   10077
$SWREG   001740              611#    1333
$SWRMK=  000200               13#      26      27    9356    9357    9378    9379
$TAB     066500            10557    11711#  14222   14249   14259   14268
$TESTN   001722              602#    9403*
$TIMES   001702              588#    1309*   1843*   2074*   2293*   2498*   2689*   2761*   2893*   3173*   3413*   3681*   3977*
                            4262*    4395*   4533*   4690*   4931*   5115*   5232*   5409*   5563*   5962*   6304*   9042*   9078*
                            9113*    9314*   9391*   9398    9401*   9411
$TKB     001546              539#    1507*   9531    9542    9559    9613    9619   10374   10410*
$TKS     001544              538#    1508*   9531    9540    9556    9580*   9611    9617   10411*
$TMP0    001632              568#    1848*   2079*   2298*   2503*   2694*   2766*   2897*   3179*   3418*   3688*   3984*   4267*
                            4400*    4538*   4695*   4870*   4936*   5120*   5237*   5414*   5568*   5714*   5967*   6309*   6764*
                            7786*   14201   14204   14212   14222   14227   14231   14234   14237   14239   14243   14253   14263
                           14272    14275   14278   14281   14284   14287   14290   14293   14296   14299   14302   14307   14310
                           14314    14319   14325   14334   14342   14351   14359   14376   14381   14398   14401   14406
$TMP1    001634              569#    2013*   2247*   2448*   2647*   3049*   3069*   3285*   3530*   4980*   4986*   4996*   5002*
                            5012*    5018*   5028*   5034*   5146*   5170*   5178*   5193*   5202*   5265*   5289*   5308*   5352*
                            5380*    5435*   5481*   5589*   5635*   5756*   5918*   5925    6022*   6037*   6051*   6066*   6125*
                            6145*    6164*   6184*   6279*   6350*   6365*   6379*   6394*   6425*   6442*   6461*   6478*   6497*
                            6514*    6533*   6550*   6577*   6600*   6623*   6646*   6668*   6672*   6681*   6686*   7045*   7213*
                            7377*    7546*   7647*   7809*   7849*   7873*   7896*  10225*  10242*  10255*  14201   14204   14222
                           14227    14231   14234   14237   14239   14243   14253   14263   14275   14281   14284   14287   14293
                           14299    14310   14398   14401   14406
$TMP10   001652              576#    3343*   3912*   4211*   4346*   4377*   4481*   4512*   4642*   4673*   4802*   4833*   5995*
                            5996*    6328*   6329*  14246   14256   14265   14328   14334   14345   14351
$TMP11   001654              577#    4347*   4378*   4482*   4513*   4643*   4674*   4803*   4834*  14328   14345
$TMP12   001656              578#    4348*   4368*   4483*   4503*   4644*   4664*   4804*   4824*  14325   14334   14342   14351
$TMP13   001660              579#    4349*   4484*   4645*   4805*
$TMP14   001662              580#    4343*   4344*   4478*   4479*   4639*   4640*   4799*   4800*  14328   14345
$TMP15   001664              581#    4345*   4480*   4641*   4801*
$TMP16   001666              582#   14325   14342
$TMP17   001670              583#    4335*   4470*   4631*   4791*
$TMP2    001636              570#    2721*   2737*   2829*   2854*   3050*   3070*   3275*   3339*   3531*   3532*   3554*   3572*
                            3791*    3863*   3905*   4091*   4164*   4204*   4370*   4505*   4666*   4826*   4904*   4981*   4982*
                            4987*    4997*   4998*   5003*   5013*   5014*   5019*   5029*   5030*   5035*   5044*   5050*   5147*
```

```
                         5172*   5179*   5194*   5203*   5266*   5267*   5290*   5309*   5354*   5434*   5463*   5480*   5509*
                         5588*   5617*   5634*   5663*   5757*   5758*   5919*   6021*   6035*   6036*   6050*   6064*   6065*
                         6124*   6143*   6144*   6163*   6182*   6183*   6278*   6349*   6363*   6364*   6378*   6392*   6393*
                         6424*   6441*   6460*   6477*   6496*   6513*   6532*   6549*   6575*   6576*   6598*   6599*   6621*
                         6622*   6644*   6645*   6675*   6679    6690*   6694    7047*   7215*   7379*   7548*   7648*   7810*
                         7850*   7874*   7897*  10226*  10243*  14201   14204   14212   14222   14231   14239   14243   14253
                        14263   14284   14287   14302   14307   14314   14319   14337   14354   14359   14376   14381   14398
                        14401   14406
$TMP20   001672           584#   4337*   4472*   4633*   4793*
$TMP21   001674           585#
$TMP22   001676           586#
$TMP23   001700           587#
$TMP3    001640           571#   2015*   2249*   2449*   2648*   2722*   2738*   2830*   2855*   3051*   3071*   3338*   3529*
                         3555*   3573*   3792*   3864*   3907*   4092*   4165*   4206*   4338*   4371*   4473*   4506*   4634*
                         4667*   4794*   4827*   5180*   5204*   5310*   5355*   5381*   5464*   5510*   5618*   5664*   5920*
                         6261*   6266*   6270*   6275*   6676*   6691*   7046*   7214*   7378*   7547*   7811*   7852*   7875*
                         7898*  10227*  10244*  14222   14227   14231   14234   14239   14243   14253   14263   14275   14281
                        14293   14299   14302   14314   14319   14325   14342   14376   14381   14398   14406
$TMP4    001642           572#   2016*   2250*   2450*   2649*   2723*   2739*   2831*   2856*   3072*   3341*   3556*   3574*
                         3793*   3865*   3908*   3915*   3920    4093*   4166*   4207*   4214*   4219    4339*   4372*   4373*
                         4474*   4507*   4508*   4635*   4668*   4669*   4795*   4828*   4829*   5181*   5205*   5311*   5357*
                         5382*   5465*   5511*   5619*   5665*   5921*   6260*   6264*   6265*   6269*   6273*   6274*   6669*
                         6673*   6682*   6683*   6687*   6688*  10241*  14204   14222   14227   14231   14234   14239   14243
                        14253   14263   14287   14319   14337   14354
$TMP5    001644           573#   2719*   2736*   2827*   2853*   3073*   3344*   3909*   4208*   4340*   4374*   4475*   4509*
                         4636*   4670*   4796*   4830*   5358*   5359*   5383*   5923*   5991*   6324*  14204   14227   14246
                        14256   14265   14302   14319   14325   14342
$TMP6    001646           574#   3345*   3910*   4209*   4334*   4341*   4375*   4469*   4476*   4510*   4630*   4637*   4671*
                         4790*   4797*   4831*   5360*   5924*   5992*   5993*   6325*   6326*  14204   14222   14246   14256
                        14265   14328   14334   14345   14351
$TMP7    001650           575#   3342*   3911*   4210*   4342*   4376*   4477*   4511*   4638*   4672*   4798*   4832*   5356*
                         5994*   6327*  14222   14246   14256   14265   14319   14337   14354
$TN    - 000061            10#   1512    1519#   1560    1570#   1603    1611#   1714    1725#   1784    1843#   1844    1845
                         2051    2074#   2075    2076    2279    2293#   2294    2295    2484    2498#   2499    2500    2679
                         2689#   2690    2691    2752    2761#   2762    2763    2870    2893#   2894    3162    3173#   3174
                         3175    3375    3413#   3414    3415    3656    3681#   3682    3683    3952    3977#   3978    3979
                         4249    4262#   4263    4264    4386    4395#   4396    4397    4520    4533#   4534    4535    4681
                         4690#   4691    4692    4842    4866#   4867    4923    4931#   4932    5107    5115#   5116    5224
                         5232#   5233    5391    5409#   5410    5411    5545    5563#   5564    5565    5700    5709#   5939
                         5962#   5963    6289    6304#   6305    6701    6760#   7772    7782#   8016    8024#   8061    8069#
                         8086    8091    8099#   8116    8125#   8146    8154#   8181    8201#   8295    8299    8315#   8374
                         8382    8404#   8498    8502    8515#   8627    8635    8645#   8669    8690#   8776    8784    8799#
                         8826    8835    8841    8848#   8953    8959    8974#   9037    9042#   9046    9073    9078#   9082
                         9108    9113#   9117    9143    9148#
$TPB     001552           541#   9783*   9794
$TPFLG   001557           545#   9732    9794
$TPS     001550           540#   9781    9794
$TRAP    054374          1304    9986#
$TRAP2   054416          9997#  10008
$TRP   - 000033         10001#  10010#  10011#  10012#  10013#  10014#  10015   10016#  10017   10018#  10019#  10020#  10021#
                        10022#  10023   10024#  10025#  10026#  10027#  10028#  10029#  10030#  10031#  10032#  10033#  10034
                        10035#  10036#  10037#  10038#  10039#
$TRPAD   054430          9991   10008#
$TSTM    001404           501#
$TSTNM   001502           518#   1291*   1848    2079    2298    2503    2694    2766    2897    3179    3418    3688    3984
                         4267    4400    4538    4695    4870    4936    5120    5237    5414    5568    5714    5967    6309
```

```
                          6764     7786     9313*    9351     9380     9402*    9403     9408     9412     9431     9469    14413
$TTYIN  053176            9638     9639     9656     9660#
$TYPBN= ****** U         10014
$TYPDS  054046            9885#   10013
$TYPE   053336            9502     9732#   10001    10009
$TYPEC  053550            9582     9762     9769     9776     9781#    9782
$TYPEX  053616            9787     9789     9792#
$TYPOC  053644            9825#   10010
$TYPON  053660            9824     9827#   10012
$TYPOS  053620            9820#   10011
$UNIT   001730             605#
$UNITM  001410             503#
$USWR   001742             612#
$XTSTR  051532            9365#
$$GET4= 000001            9329#    9331#
$OFILL  054043            9821*    9825*    9835     9870#
$40CAT= ****** U          9362     9441
.     = 120316             461#     465#     475      476#     478#     480#     482#     488      489#     491#     493#     515#     594
                          1297     1312     1313     1355#    1372#    1530     1533#    1578     1581#    1662     1665#    1689     1692#
                          1757     1760#    3119#    4885     4889     5448     5451#    5497     5500#    5602     5605#    5651     5654#
                          6664#    7918#    8204     8407     9341     9345     9411     9412     9469     9523#    9531     9660#    9661
                          9667     9794     9939#   10055    10079    10220#   10721#   11691#   14197#   14415    14418#   14419#
.$ASTA- ****** U          9475     9478
.$X   = 001400             488#     493
```

```
ADDDPA      1#    1908    2355    3137    3464    3740    3813    4039    4114
ADJUST      1#    5447    5496    5601    5650
BYTLT1      1#    4327    4462    4623    4783
BYTLT2      1#    4367    4502    4663    4823
CHAIN       1#    9329
CLRMAC      1#   10267
CLRRFL      1#
CMPDPA      1#    1887    1919    1988    2136    2155    2223    2335    2366    2422    2549    2567    2621    2987    3015
          3237    3313    3477    3621    3751    3824    4050    4125
CNVDPA      1#    1939    2175
CNVMAP      1#    2386    2587
COMMEN      1#     413#
DD          1#    6892    7060    7228    7393    7561
DRIVER      1#    6895    7063    7231    7396
ENDCOM      1#     413#
ERCLR       1#    9466
ERROR      36#    1540    1545    1549    1553    1589    1594    1630    1647    1676    1704    1769    1776    1985    2019
          2220    2253    2418    2453    2618    2652    2724    2740    2832    2857    3052    3074    3276    3346    3533
          3557    3575    3794    3866    3913    4094    4167    4212    4350    4379    4485    4514    4646    4675    4806
          4835    4905    4983    4988    4999    5004    5015    5020    5031    5036    5045    5051    5148    5173    5182
          5195    5206    5268    5291    5361    5385    5436    5466    5482    5512    5590    5620    5636    5666    5759
          5927    5931    5933    6023    6038    6052    6067    6126    6146    6165    6185    6282    6351    6366    6380
          6395    6426    6443    6462    6479    6498    6515    6534    6551    6578    6601    6624    6647    6678    6693
          7048    7216    7380    7549    7649    7814    7818    7820    7854    7877    7900    7967    7985    8003    8028
          8034    8039    8045    8052    8058    8074    8082    8088    8105    8111    8132    8140    8162    8174    8230
          8246    8282    8354    8362    8433    8449    8485    8563    8609    8659    8731    8757    8816    8831    8881
          8896    8903    8922    8933    8941    8999    9008    9017    9053    9060    9067    9089    9096    9103    9124
          9131    9138    9226    9247    9279   10229   10246   10938   11098   11256   11442   11569
ESCAPE      1#     413#
EXTRA       1#     622
GETPRI      1#     413#  10117
GETSWR      1#     413#   1342#
HANREG      1#   10916   11076   11234   11420   11547
IIMAC1      1#    5391    5545
ITEMAC      1#    1092
JJM1        1#    4975    4991    5007    5023
KB       8007#    8021    8066    8096    8122    8151    8198    8312    8401    8512    8642    8687    8796    8845    8971
KKM1        1#    6078    6082    6086    6090    6097    6101    6105    6109
KKM2        1#    6114    6133    6153    6172
KMAC1       1#
KMAC5       1#
MMAC1       1#
MMAC2       1#
MMAC3       1#
MMAC4       1#
MMAC5       1#
MMM1        1#    6419    6436    6455    6472    6491    6508    6527    6544
MMM2        1#    6566    6588    6611    6634
MSG      1784#    1786    2050#   2053    2679#   2681    2869#   2872    3161#   3164    3375#   3377    3655#   3658    3951#
          3954    4841#   4844    4923#   4925    5107#   5109    5224#   5226    5391#   5393    5545#   5547    5699#   5702
          5939#   5941    6288#   6291    7771#   7774    8016#   8018    8061#   8063    8091#   8093    8116#   8118    8146#
          8148    8181#   8183    8299#   8301    8382#   8384    8501#   8504    8635#   8637    8668#   8671    8783#   8786
          8841#   8843    8959#   8961    9035#   9039    9073#   9075    9108#   9110    9143#   9145
MSGS     6700#    6703
MSG1     2278#    2281    2486
MSG167   1512#    1514
```

```
MSG170    1560#    1562
MSG171    1603#    1605
MSG172    1714#    1716
MSG2      2751#    2754
MSG201    5390#    5393    5547
MSG3      3654#    3658    3954    4248#    4251    4522
MSG300   13063#   13105   13147   13189
MSG301   13231#   13271   13312   13353
MSG4      4385#    4388    4683
MULT         1#     413#
NEWTST       1#     413#   1512    1560    1603    1714    1784    2051    2279    2484    2679    2752    2870    3162    3375
          3656     3952    4249    4386    4520    4681    4842    4923    5107    5224    5391    5545    5700    5939    6289
          6701     7772    8016    8061    8091    8116    8146    8181    8299    8382    8502    8635    8669    8784    8841
          8959     9037    9073    9108    9143
NMAC1        1#
NMAC2        1#
NMAC3        1#
POP          1#     413#   9517    9518    9707    9926    9971   10064   10065
PUSH         1#     413#   9478    9480    9501    9687    9885    9952   10045   10051
QQM1         1#    5836    5856    5876
REPORT       1#     413#
SCOPE       37#    1518    1569    1610    1724    1842    2073    2292    2497    2688    2760    2892    3172    3412    3680
          3976     4261    4394    4532    4689    4865    4930    5114    5231    5408    5562    5708    5961    6303    6759
          7781     8023    8068    8098    8124    8153    8200    8314    8403    8514    8644    8689    8798    8847    8973
          9041     9077    9112    9147    9312
SETMAP       1#    4541    4701
SETMM        1#    2303    2506    2780    2905    3212    3421    3714    4012    6765
SETPRI       1#     413#
SETTRA   10001#   10010   10011   10012   10013   10015   10017   10018   10019   10020   10021   10023   10024   10025   10026
         10027    10028   10029   10030   10031   10032   10034   10035   10036   10037   10038
SETUP        1#     413#   1292
SKBREX       1#    5416    5570
SKIP         1#     413#   8086    8295    8374    8496    8627    8776    8826    8835    8953    9046    9082    9117
SLASH        1#     413#
SPACE      413#
SSKAD        1#    1845    2076    2295    2500    2691    2763    2894    3175    3415    3683    3979    4264    4397    4535
          4692     4867    4932    5116    5233    5411    5565    5709    5963    6305    6760    7782
STARS        1#     413#    473     485     487     494     511     594     597    1374    1512    1517    1560    1568    1603
          1609     1615    1632    1649    1677    1711    1714    1723    1784    1841    2051    2072    2279    2291    2484
          2496     2679    2687    2752    2759    2870    2891    3162    3171    3375    3411    3656    3679    3952    3975
          4249     4260    4386    4393    4520    4531    4681    4688    4842    4864    4923    4929    5107    5113    5224
          5230     5391    5407    5545    5561    5700    5707    5939    5960    6289    6302    6701    6758    7772    7780
          8016     8022    8061    8067    8091    8097    8116    8123    8146    8152    8181    8199    8299    8313    8382
          8402     8502    8513    8635    8643    8669    8688    8784    8797    8841    8846    8959    8972    9037    9040
          9073     9076    9108    9111    9143    9146    9304    9348    9415    9473    9530    9533    9601    9630    9671
          9717     9797    9875    9943    9980   10041   10057   10083   10184   10853   10911
SWRSU        1#     413#   1314#
THIT         1#    5429    5475    5583    5629    6016    6030    6045    6059    6119    6138    6158    6177    6344    6358
          6373     6387    6570    6593    6616    6639
TOMAP3       1#    2801    2816    2840
TOPAR3       1#    3034    3054    3258    3776    3849    4076    4150
TRMTRP   10001#
TSTHIT       1#    6014    6027    6042    6056    6342    6355    6370    6384    6568    6591    6614    6637
TYPBIN       1#     413#
TYPDEC       1#     413#   9324
TYPNAM       1#     413#   1335
```

```
TYPNUM       1#    413#
TYPOCS       1#    413#   1389    1394
TYPOCT       1#    413#   9551
TYPTXT       1#    413#
UMAC1        1#
UMAC2        1#   5518    5528    5672    5682
UMAC3        1#   5461    5507    5615    5661
$$CMRE     509#    548     549     550     551     552     553     554     555     556     557     558     559     560     561
           562     563     564     565     566     567
$$CMTM     509#    568     569     570     571     572     573     574     575     576     577     578     579     580     581
           582     583     584     585     586     587
$$ESCA       1#    413#
$$NEWT       1#    413#   1512    1560    1603    1714    1784    2051    2279    2484    2679    2752    2870    3162    3375
          3656    3952    4249    4386    4520    4681    4842    4923    5107    5224    5391    5545    5700    5939    6289
          6701    7772    8016    8061    8091    8116    8146    8181    8299    8382    8502    8635    8669    8784    8841
          8959    9037    9073    9108    9143
$$SET    10001#  10010   10011   10012   10013   10015   10017   10018   10019   10020   10021   10023   10024   10025   10026
         10027   10028   10029   10030   10031   10032   10034   10035   10036   10037   10038
$$SETM    1330#
$$SKIP       1#    413#   8086    8295    8374    8498    8627    8776    8826    8835    8953    9046    9082    9117
.EQUAT       1#
.HEADE       1#
.KT11        1#
.SETUP       1#    442
.SWRHI       1#     15
.SWRLO      27#
.$ACT1       1#    471
.$APTB       1#    595#
.$APTH       1#    483
.$APTY       1#   9471
.$ASTA       1#
.$CATC       1#    459
.$CMTA       1#    509
.$DB2D       1#
.$DB20       1#  10182
.$DIV        1#
.$EOP        1#   9302
.$ERRO       1#   9413
.$ERRT       1#
.$MULT       1#
.$POWE       1#  10039
.$RAND       1#   9941
.$RDDE       1#
.$RDOC       1#
.$READ       1#   9528
.$R2AZ       1#
.$SAVE       1#   9669
.$SB2D       1#
.$SB20       1#
.$SCOP       1#   9346
.$SIZE       1#  10081
.$SUPR       1#
.$TRAP       1#   9978
.$TYPB       1#
.$TYPD       1#   9873
.$TYPE       1#   9715
```

```
.$TYPO       1#   9795
.$40CA       1#
.1170        1#     29


. ABS.  120316     000


 ERRORS DETECTED:  0

 CEKBDE.BIN,CEKBDE.LST/CRF/SOL=CEKBDE.SML,CEKBDE.P11
 RUN-TIME: 100 130 19 SECONDS
 RUN-TIME RATIO: 621/250=2.4
 CORE USED:  45K  (89 PAGES)
```