PDP11/70

CACHE DIAGNOSTIC PART 1

MD-11-DEKBC-B

EP-DEKBC-DL-A

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DEKBC-B-D

PRODUCT NAME: PDP-11/70 CACHE DIAGNOSTIC PART 1

DATE CREATED: 11-SEPT-75

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: ANTHONY VEZZA

CONTENTS

1. ABSTRACT

THE PROGRAMS, DEKBC AND DEKBD, ARE INTENDED TO BE USED AS AIDS FOR THE REPAIR AND MAINTENANCE OF THE CACHE MEMORY SYSTEM IN THE PDP 11/70 COMPUTING SYSTEM. THE AIM IS TO DETECT AND REPORT FAILING COMPONENTS OF THE CACHE UNIT. THE FAILURES ARE TYPICALLY IDENTIFIED WITH A FAILING CIRCUIT WHEN THE REPORT IS MADE, BUT THE OVERALL DIAGNOSTIC PHILOSOPHY HAS BEEN TO LOCATE THE FAILING MODULE (HEX BOARD) OF WHICH THERE ARE FOUR (4) IN THE CACHE UNIT. NOTE THAT WHEN IS FAILURE IS REPORTED AND THE ASSOCIATED CIRCUIT IDENTIFIED, THAT CIRCUIT SHOULD NOT BE TAKEN IN BLIND FAITH 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, DEKBC, IS DESIGNED TO TEST THE FIRST TWO OF THESE BOARDS; THE PROGRAM, DEKBD, IS DESIGNED TO TEST THE LAST TWO BOARDS. NOTE THAT THOUGH THE TESTING HAS BEEN DIVIDED INTO TWO STAND ALONE PROGRAMS EACH ASSOCIATED WITH TWO MODULES IT SHOULD NOT BE ASSUMED THAT A PARTICULAR MODULE IS WORKING AFTER HAVING RUN ONLY ONE OF THE PROGRAMS! BOTH PROGRAMS SHOULD BE RUN! FOR EXAMPLE, JUST RUNNING DEKBC WITHOUT ERROR DOES NOT RULE OUT A FAILTY COMPONENT ON THE CCB (CACHE CONTROL) BOARD. TO PUT IT MORE SIMPLY THE TESTING HAS BEEN DIVIDED INTO TO PROGRAMS ONLY BECAUSE OF THE RESTRICTIONS OF CORE SIZE! AND NOT TO PROVIDE A MEANS OF TESTING TWO OF THE BOARDS WITH ONE PROGRAM AND THE OTHER TWO BOARDS WITH A SECOND PROGRAM. NOTE THAT DEKBD IS DESIGNED TO RUN AFTER DEKBC. IF THIS HIERARCHY IS NOT NEEDED, THAT IS IF DEKBD IS RUN BEFORE DEKBC, THEN THE ERROR REPORTING FROM DEKBD SHOULD NOT BE STRICTLY INTERPRETED.

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

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

2.3 PRELIMINARY PROGRAMS

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

3. LOADING PROCEDURE

3.1 METHOD (TO BE SUPPLIED)

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS (SEE 5.1)

4.2 STARTING ADDRESS 200

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

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

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

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS FOR DEKBC:

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

DEKBD USES THE SAME SWITCH SETTINGS AS DEKBC EXCEPT

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

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

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

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

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

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

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

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

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

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

6. ERRORS

6.1 ERROR HALTS ONLY TEST NUMBER 14 IN PROGRAM DEKBC, THE MAINTENANCE REGISTER COUNT PATTERN TEST, HALTS THE PROCESSOR IN THE SITUATION WHERE IT CAN'↑ CLEAR THE MAINTENANCE REGISTER. HERE PROCEDING WITH

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

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

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS NONE

7.2 OPERATING RESTRICTIONS THE MONITOR OR LOADER (OR WHAT EVER IS IN THE FIRST 28K OF MEMORY FROM LOCATIONS 152000 THROUGH LOCATION 157776 ARE SAVED SO THAT THE USER CAN RESTORE HIS LOADER OR MONITOR BY TYPING CONTROL-C (↑C) , (SEE 4.3 AND 5.2.7). IF THE PROGRAM WAS CHAINED IN BY A MONITOR WHICH WANTS CONTROL AUTOMATICALLY PASSED BACK TO IT WHEN TESTING IS DONE THAT MONITOR IS RESTORED AND CONTROL IS GIVEN TO IT BY THE END OF PASS 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 LOCATIO 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 OVERIDED BY A ONE (1) MAKING ITERATIONS MEANINGLESS ON THAT FIRST PASS.

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

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

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

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

8.9 CRITICAL DEPENDENCE OF SOME TESTS ON THE

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

9. PROGRAM DESCRIPTION

9.1 DEKBC

PROGRAM BY ANTHONY S. VEZZA

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

TEST 1 CACHE REGISTERS RESPONSE TEST

REFERENCE EACH CACHE REGISTER MAKING SURE SUCH REFERENCES DO NOT TIME OUT.

TEST 2 CACHE REGISTERS DATA PATH, READ ZEROES TEST

THIS TEST CHECKS THE ABILITY OF THE CACHE REGISTER DATA PATHS TO PASS 0'S BY FIRST WRITING THEN READING 0'S AT THE CONTROL AND MAINTENANCE REGISTERS.

TEST 3 CACHE REGISTERS DATA PATH, READ ONES TEST

THIS TEST PERFORMS A READ OF BOTH THE HIGH ORDER AND LOW ORDER ERROR

ADDRESS REGISTER. THIS IS DONE TO MAKE SURE THAT THE REGISTERS' DATA PATHS CAN PASS ONES. NOTE THAT THE LOW ORDER ADDRESS REGISTER SHOULD CONTAIN A 177740 AND THE HIGH ORDER REGISTER SHOULD CONTAIN 000003; THIS LEAVES THE DATA PATH LINE'S BITS 2,3 AND 4 UNTESTED FOR THEIR AVAILITY TO PASS ONES. THIS WILL BE CHECKED IN THE COUNT PATTERN TST4.

TEST 4 CACHE CONTROL REGISTER COUNT PATTERN TEST

THIS TEST RUNS A COUNT PATTERN THROUGH THE CACHE CONTROL REGISTER FOR THE PURPOSE OF CHECKING OUT THE DATA RELIABILITY OF BOTH THE REGISTER BITS AND THE DATA PATHS LINES.

TEST 5 CACHE HIT/MISS AND CONTROL REGISTER SIMPLE MISSES TEST

THIS IS A TEST OF THE HIT/MISS REGISTER AND THE CONTRL REGISTER'S ABILITY TO FORCE MISSES. ZEROES ARE FLOATED THROUGH THE HIT/MISS REGISTER.

TEST 6 CACHE HIT/MISS AND CONTROL REGISTER SIMPLE HIT TEST

THIS IS A TEST OF THE HIT/MISS REGISTER AND THE THE FORCE MISS BITS OF THE CONTROL REGISTER. WHAT IS DONE IS TO SEE IF ANY HITS AT ALL ARE POSSIBLE WITH THE CONTROL REGISTER CLEARED. THEN THE SAME IS DONE WITH EACH GROUP DISABLE ONE AT A TIME. BY DISABLED IS MEANT THAT THE FORCE MISS BIT IS SET IN THE CONTROL REGISTER FOR THE DISABLED GROUP AND THE FORCE SELECT BIT IS SET FOR THE OTHER GROUP.

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

THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS OF FORCE MISS AND

FORCE SELECTION. AN ADDRESS IS MADE A HIT IN GROUP ONE; THEN ANOTHER ADDRESS, WHOSE HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING SELECTION OF GROUP ZERO; THEN SEE IF THE FIRST ADDRESS IS STILL A HIT IN GROUP ONE; FINALLY TURN ON THE FORCE MISS GROUP ZERO BIT AND SEE IF THE SECOND ADDRESS' HIT IN GROUP ZERO CAN BE FORCED TO A MISS.

TEST 10 CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 1 TEST

THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS MADE A HIT IN GROUP ZERO; THEN ANOTHER ADDRESS, WHOSE HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING SELECTION OF GROUP ONE; THEN SEE IF THE FIRST ADDRESS IS STILL A HIT IN GROUP ZERO; FINALLY TURN ON THE FORCE MISS GROUP ONE BIT AND SEE IF THE SECOND ADDRESS' HIT IN GROUP ONE CAN BE FORCED TO A MISS.

TEST 11 CACHE HIT/MISS REGISTER PATTERNS TEST

THIS IS A TEST OF THE HIT/MISS REGISTER WHICH FLOATS DIFFERENT PATTERNS OF HITS AND MISSES THROUGH THAT REGISTER. THIS IS DONE FIRST WITH BOTH GROUPS ENABLE; THEN WITH GROUP ZERO DISABLED THAT IS FORCING SELECTION OF GROUP ONE AND FORCING MISSES TO GROUP ZERO; FINALLY WITH GROUP ONE DISABLED.

TEST 12 CACHE CONTROL AND HIT/MISS REGISTERS EVALUATION ROUTINE

THIS IS NOT A TEST. THIS ROUTINE IS USED TO LOOK AT THE RESULTS OF TST5 THROUGH TST10, WHICH TESTED THE HIT/MISS REGISTER AND THE CONTROL REGISTER. THOSE TESTS HAVE

SIGNALLED A BAD REGISTER USING THE FLAGS, CONFL2 AND HIMFL2, REPRESENTING THE CONTROL AND HIT/MISS REGISTERS RESPECTIVELY. IF ONE OF THESE REGISTERS WAS FOUND TO BE BAD THE FLAG SHOULD BE A -1. WHILE A ZERO FLAG INDICATES THAT THOSE TESTS FOUND THAT REGISTER FUNCTIONAL. THIS ROUTINE LOOKS AT THE FLAGS, CONFL2 AND HIMFL2, WHICH ARE CONSIDERED TO BE LOCAL AND TRANSFERS THE INDICATORS THEY CONTAIN TO THE GLOBAL FLAGS, CONFLG AND HIMFLG. THESE GLOBAL FLAGS ARE USED TO DESIGNATE TO THE REST OF THE PROGRAM THE FUNCTIONALITY OR DISFUNCTIONALITY OF THOSE REGISTERS.

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

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

TEST 14 CACHE MAINTENANCE REGISTER CCUNT PATTERN TEST

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

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

TEST 15 CACHE MAINTENANCE AND ERROR REGISTERS TEST 1

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

TEST 16 CACHE MAINTENANCE AND ERROR REGISTERS TEST 2

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

TEST 17 CACHE MAINTENANCE AND ERROR REGISTERS TEST 3

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

TEST 20 CACHE MAINTENANCE AND ERROR REGISTERS TEST 4

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

TEST 21 CACHE MAINTENANCE AND ERROR REGISTERS TEST 5

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

TEST 22 CACHE MAINTENANCE AND ERROR REGISTERS TEST 6

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

TEST 23 CACHE MAINTENANCE AND ERROR REGISTERS TEST 7

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

TEST 24 CACHE MAINTENANCE AND ERROR REGISTERS TEST 10

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

TEST 25 CACHE MAINTENANCE AND ERROR REGISTERS TEST 11

THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO, FOR THE HIGH BYTE OF THE

ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S ABILITY TO SET CORRECTLY FOR THIS ERROR. THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU TO THE CACHE.

TEST 26 CACHE MAINTENANCE AND ERROR REGISTERS TEST 12

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

TEST 27 CACHE MAINTENANCE AND ERROR REGISTERS TEST 13

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

TEST 30 CACHE MAINTENANCE AND ERROR REGISTERS TEST 14

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

TEST 31 CACHE MAINTENANCE AND ERROR REGISTERS TEST 15

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

TEST 32 CACHE MAINTENANCE AND ERROR REGISTERS TEST 16

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

TEST 33 CACHE MAINTENANCE AND ERROR REGISTERS TEST 17

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

TEST 34 CACHE MAINTENANCE AND ERROR REGISTERS TEST 20

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO MAKE THAT REFERENCE CAUSE A MAIN MEMORY ADDRESS AND CONTROL LINES PARITY ERROR ON THE MAIN MEMORY BUS.

TEST 35 CACHE MAINTENANCE AND ERROR

REGISTERS TEST 21

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA PARITY ERROR ON THAT REFERENCE WHICH IS TO AN EVEN WORD IN THE PAIR, WHICH IS ALSO THE WANTED WORD.

TEST 36 CACHE MAINTENANCE AND ERROR REGISTERS TEST 22

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA PARITY ERROR ON THAT REFERENCE WHICH IS TO AN ODD WORD IN THE PAIR, WHICH IS ALSO THE WANTED WORD.

TEST 37 CACHE MAINTENANCE AND ERROR REGISTERS TEST 23

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT ADDRESS .

TEST 40 CACHE MAINTENANCE AND ERROR REGISTERS TEST 24

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND

THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT ADDRESS .

TEST 41 CACHE MAINTENANCE AND ERROR REGISTERS TEST 25

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT DATA .

TEST 42 CACHE MAINTENANCE AND ERROR REGISTERS TEST 26

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE. THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE LOW BYTE OF THAT DATA .

TEST 43 CACHE ERROR REGISTER UNIBUS TIME OUT TEST

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO COMPREHEND A CPU TO UNIBUS THROUGH THE MAP TO THE CACHE REFERENCE WHICH TIMES OUT IN MAIN MEMORY. MANY SUCH NON-EXISTENT MEMORY LOCATIONS ARE CONVIENTLY GUARENTEED TO EXIST! ALL THE ADDRESSES FROM 17000000 THROUGH 17777776 ARE ADDRESSES WHICH CAN NOT EXIST. HERE ONLY ONE OF THESE ADDRESSES, 17777776, WILL BE USED TO CAUSE A TIME OUT ON THE UNIBUS AN

THE CONSEQUENT ABORT TO VECTOR ERRVEC.

TEST 44 CACHE CONTROL REGISTER DISABLE TRAPS TEST 1

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

TEST 45 CACHE CONTROL REGISTER DISABLE TRAPS TEST 2

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

TEST 46 CACHE CONTROL REGISTER DISABLE TRAPS TEST 3

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

TEST 47 CACHE ERROR REGISTER LOCK UP TEST 1

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

TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE CACHE DIRECTLY. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE CACHE DIRECTLY.

TEST 50 CACHE ERROR REGISTER LOCK UP TEST 2

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE CACHE DIRECTLY. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.

TEST 51 CACHE ERROR REGISTER LOCK UP TEST 3

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO

THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE CACHE DIRECTLY.

TEST 52 CACHE ERROR REGISTER LOCK UP TEST 4

THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE. THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.

TEST 53 MAIN MEMORY DATA PARITY CHECKERS LOW BYTE TEST

THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS FOR THE LOW BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD. THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1). THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA PARITY CHECKERS WORKS IN SUCH A WAY AS TO EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS ALREADY ONE THEN NO ERROR OCCURS!

TEST 54 MAIN MEMORY DATA PARITY CHECKERS HIGH BYTE TEST

THIS IS A TEST OF THE TWO MAIN

MEMORY DATA PARITY CHECKERS FOR THE HIGH BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD. THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1). THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA

PARITY CHECKERS WORKS IN SUCH A WAY AS TO EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS ALREADY ONE THEN NO ERROR OCCURS!

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

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

```
                                ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
167400                          $SWR=167400
000200                          $SWRMK=200

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

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100            ;;FIRST ADDRESS OF THE STACK
001100                          KERSTK= STACK           ;;KERNEL STACK
000700                          SUPSTK= STACK-200       ;;SUPERVISOR STACK
000600                          USESTK= STACK-300       ;;USER STACK
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          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 LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

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

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

```
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
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
                170210          MAPL02 = 170210
                170212          MAPH02 = 170212
                170214          MAPL03 = 170214
                170216          MAPH03 = 170216
                170220          MAPL04 = 170220
                170222          MAPH04 = 170222
                170224          MAPL05 = 170224
                170226          MAPH05 = 170226
```

```
                170230                  MAPL06 = 170230
                170232                  MAPH06 = 170232
                170234                  MAPL07 = 170234
                170236                  MAPH07 = 170236
                170240                  MAPL10 = 170240
                170242                  MAPH10 = 170242
                170244                  MAPL11 = 170244
                170246                  MAPH11 = 170246
                170250                  MAPL12 = 170250
                170252                  MAPH12 = 170252
                170254                  MAPL13 = 170254
                170256                  MAPH13 = 170256
                170260                  MAPL14 = 170260
                170262                  MAPH14 = 170262
                170264                  MAPL15 = 170264
                170266                  MAPH15 = 170266
                170270                  MAPL16 = 170270
                170272                  MAPH16 = 170272
                170274                  MAPL17 = 170274
                170276                  MAPH17 = 170276
                170300                  MAPL20 = 170300
                170302                  MAPH20 = 170302
                170304                  MAPL21 = 170304
                170306                  MAPH21 = 170306
                170310                  MAPL22 = 170310
                170312                  MAPH22 = 170312
                170314                  MAPL23 = 170314
                170316                  MAPH23 = 170316
                170320                  MAPL24 = 170320
                170320                  MAPH24 = 170320
                170324                  MAPL25 = 170324
                170326                  MAPH25 = 170326
                170330                  MAPL26 = 170330
                170332                  MAPH26 = 170332
                170334                  MAPL27 = 170334
                170336                  MAPH27 = 170336
                170340                  MAPL30 = 170340
                170342                  MAPH30 = 170342
                170344                  MAPL31 = 170344
                170346                  MAPH31 = 170346
                170350                  MAPL32 = 170350
                170352                  MAPH32 = 170352
                170354                  MAPL33 = 170354
                170356                  MAPH33 = 170356
                170360                  MAPL34 = 170360
                170362                  MAPH34 = 170362
                170364                  MAPL35 = 170364
                170366                  MAPH35 = 170366
                170370                  MAPL36 = 170370
                170372                  MAPH36 = 170372
                170374                  MAPL37 = 170374
                170376                  MAPH37 = 170376
                                        .EQUIV  MAPL00,MAPL0
                                        .EQUIV  MAPH00,MAPH0
                                        .EQUIV  MAPL01,MAPL1
                                        .EQUIV  MAPH01,MAPH1
```

```
                                  .EQUIV  MAPL02,MAPL2
                                  .EQUIV  MAPH02,MAPH2
                                  .EQUIV  MAPL03,MAPL3
                                  .EQUIV  MAPH03,MAPH3
                                  .EQUIV  MAPL04,MAPL4
                                  .EQUIV  MAPH04,MAPH4
                                  .EQUIV  MAPL05,MAPL5
                                  .EQUIV  MAPH05,MAPH5
                                  .EQUIV  MAPL06,MAPL6
                                  .EQUIV  MAPH06,MAPH6
                                  .EQUIV  MAPL07,MAPL7
                                  .EQUIV  MAPH07,MAPH7

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

        000000                            .=0
                                  ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                  ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                  ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
        000174                            .=174
000174  000000                    DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                    SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                  .SBTTL  STARTING ADDRESS(ES)
000200  000137  003016                    JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

                                  .SBTTL  ACT11 HOOKS

                                  ;;*****************************************************************
                                  ;HOOKS REQUIRED BY ACT11
        000204                            $SVPC=.                 ;SAVE PC
        000046                            .=46
000046  026424                            $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                            .=52
000052  000000                            .WORD   0               ;;2)SET LOC.52 TO ZERO
        000204                            .=$SVPC                 ;; RESTORE PC
```

```
                                .SBTTL  COMMON TAGS

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

                    001100              .=1100
001100              $CMTAG:                         ;;START OF COMMON TAGS
001100  000000      $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102     000      $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000      $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000      $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000      $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000      $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000      $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000      $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001      $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000      $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000      $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000      $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000      $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000      $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000              .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000              .WORD   0
001134     000      $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000      $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000              .WORD   0
001140  177570      SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570      DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
                    $TKS:   177560                  ;;TTY KBD STATUS
                    $TKB:   177562                  ;;TTY KBD BUFFER
                    $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
                    $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000      $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002      $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012      $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000      $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000      $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                    ;;WHICH  ($REG0) WAS OBTAINED
001162  000000      $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000      $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000      $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000      $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000      $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000      $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000      $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000      $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000      $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001204  000000      $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
001206  000000      $REG12: .WORD   0               ;;CONTAINS (($REGAD)+24)
001210  000000      $REG13: .WORD   0               ;;CONTAINS (($REGAD)+26)
001212  000000      $REG14: .WORD   0               ;;CONTAINS (($REGAD)+30)
001214  000000      $REG15: .WORD   0               ;;CONTAINS (($REGAD)+32)
001216  000000      $REG16: .WORD   0               ;;CONTAINS (($REGAD)+34)
001220  000000      $REG17: .WORD   0               ;;CONTAINS (($REGAD)+36)
001222  000000      $REG20: .WORD   0               ;;CONTAINS (($REGAD)+40)
```

```
    557  001224  000000                $REG21: .WORD   0               ;;CONTAINS (($REGAD)+42)
    558  001226  000000                $REG22: .WORD   0               ;;CONTAINS (($REGAD)+44)
    559  001230  000000                $REG23: .WORD   0               ;;CONTAINS (($REGAD)+46)
    560  001232  000000                $TMP0:  .WORD   0               ;;USER DEFINED
    561  001234  000000                $TMP1:  .WORD   0               ;;USER DEFINED
    562  001236  000000                $TMP2:  .WORD   0               ;;USER DEFINED
    563  001240  000000                $TMP3:  .WORD   0               ;;USER DEFINED
    564  001242  000000                $TMP4:  .WORD   0               ;;USER DEFINED
    565  001244  000000                $TMP5:  .WORD   0               ;;USER DEFINED
    566  001246  000000                $TMP6:  .WORD   0               ;;USER DEFINED
    567  001250  000000                $TMP7:  .WORD   0               ;;USER DEFINED
    568  001252  000000                $TMP10: .WORD   0               ;;USER DEFINED
    569  001254  000000                $TMP11: .WORD   0               ;;USER DEFINED
    570  001256  000000                $TMP12: .WORD   0               ;;USER DEFINED
    571  001260  000000                $TMP13: .WORD   0               ;;USER DEFINED
    572  001262  000000                $TMP14: .WORD   0               ;;USER DEFINED
    573  001264  000000                $TMP15: .WORD   0               ;;USER DEFINED
    574  001266  000000                $TMP16: .WORD   0               ;;USER DEFINED
    575  001270  000000                $TMP17: .WORD   0               ;;USER DEFINED
    576  001272  000000                $TMP20: .WORD   0               ;;USER DEFINED
    577  001274  000000                $TMP21: .WORD   0               ;;USER DEFINED
    578  001276  000000                $TMP22: .WORD   0               ;;USER DEFINED
    579  001300  000000                $TMP23: .WORD   0               ;;USER DEFINED
    580  001302  000000                $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
    581  001304  000000                $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
    582  001306  177607  000377        $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
    583  001312     077                $QUES:  .ASCII  /?/             ;;QUESTION MARK
    584  001313     015                $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
    585  001314  000012                $LF:    .ASCIZ  <12>            ;;LINE FEED
    586                                ;;**************************************************************
```

```
                                .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


001316                          $ERRTB:


                                ;ERROR TABLE FOR ERROR TYPE OUT:
                                ;ITEM 1
001316  035105  046456  050602          .WORD   EM1,DH1,DT1,DF1
001324  050405
                                ;ITEM 0
001326  000000  000000  000000          .WORD   0,0,0,0
001334  000000
                                ;ITEM 0
001336  000000  000000  000000          .WORD   0,0,0,0
001344  000000
                                ;ITEM 0
001346  000000  000000  000000          .WORD   0,0,0,0
001354  000000
                                ;ITEM 0
001356  000000  000000  000000          .WORD   0,0,0,0
001364  000000
                                ;ITEM 0
001366  000000  000000  000000          .WORD   0,0,0,0
001374  000000
                                ;ITEM 0
001376  000000  000000  000000          .WORD   0,0,0,0
001404  000000
                                ;ITEM 0
001406  000000  000000  000000          .WORD   0,0,0,0
001414  000000
                                ;ITEM 0
001416  000000  000000  000000          .WORD   0,0,0,0
001424  000000
                                ;ITEM 0
001426  000000  000000  000000          .WORD   0,0,0,0
001434  000000
                                ;ITEM 0
001436  000000  000000  000000          .WORD   0,0,0,0
001444  000000
                                ;ITEM 14
001446  035172  046531  050614          .WORD   EM14,DH14,DT14,DF14
001454  050411
                                ;ITEM 15
```

```
643  001456  035231  046624  050630          .WORD   EM15,DH15,DT15,DF15
644  001464  050416
645                                  ;ITEM 0
646  001466  000000  000000  000000          .WORD   0,0,0,0
647  001474  000000
648                                  ;ITEM 0
649  001476  000000  000000  000000          .WORD   0,0,0,0
650  001504  000000
651                                  ;ITEM 0
652  001506  000000  000000  000000          .WORD   0,0,0,0
653  001514  000000
654                                  ;ITEM 0
655  001516  000000  000000  000000          .WORD   0,0,0,0
656  001524  000000
657                                  ;ITEM 0
658  001526  000000  000000  000000          .WORD   0,0,0,0
659  001534  000000
660                                  ;ITEM 0
661  001536  000000  000000  000000          .WORD   0,0,0,0
662  001544  000000
663                                  ;ITEM 0
664  001546  000000  000000  000000          .WORD   0,0,0,0
665  001554  000000
666                                  ;ITEM 0
667  001556  000000  000000  000000          .WORD   0,0,0,0
668  001564  000000
669                                  ;ITEM 0
670  001566  000000  000000  000000          .WORD   0,0,0,0
671  001574  000000
672                                  ;ITEM 0
673  001576  000000  000000  000000          .WORD   0,0,0,0
674  001604  000000
675                                  ;ITEM 0
676  001606  000000  000000  000000          .WORD   0,0,0,0
677  001614  000000
678
679                                  ;ITEM 0
680  001616  000000  000000  000000          .WORD   0,0,0,0
681  001624  000000
682                                  ;ITEM 0
683  001626  000000  000000  000000          .WORD   0,0,0,0
684  001634  000000
685                                  ;ITEM 0
686  001636  000000  000000  000000          .WORD   0,0,0,0
687  001644  000000
688                                  ;ITEM 0
689  001646  000000  000000  000000          .WORD   0,0,0,0
690  001654  000000
691                                  ;ITEM 0
692  001656  000000  000000  000000          .WORD   0,0,0,0
693  001664  000000
694                                  ;ITEM 0
695  001666  000000  000000  000000          .WORD   0,0,0,0
696  001674  000000
697                                  ;ITEM 0
698  001676  000000  000000  000000          .WORD   0,0,0,0
```

```
  699  001704  000000
  700                                  ;ITEM 0
  701  001706  000000  000000  000000          .WORD   0,0,0,0
  702  001714  000000
  703                                  ;ITEM 0
  704  001716  000000  000000  000000          .WORD   0,0,0,0
  705  001724  000000
  706                                  ;ITEM 0
  707  001726  000000  000000  000000          .WORD   0,0,0,0
  708  001734  000000
  709                                  ;ITEM 0
  710  001736  000000  000000  000000          .WORD   0,0,0,0
  711  001744  000000
  712                                  ;ITEM 0
  713  001746  000000  000000  000000          .WORD   0,0,0,0
  714  001754  000000
  715                                  ;ITEM 0
  716  001756  000000  000000  000000          .WORD   0,0,0,0
  717  001764  000000
  718                                  ;ITEM 0
  719  001766  000000  000000  000000          .WORD   0,0,0,0
  720  001774  000000
  721                                  ;ITEM 0
  722  001776  000000  000000  000000          .WORD   0,0,0,0
  723  002004  000000
  724                                  ;ITEM 0
  725  002006  000000  000000  000000          .WORD   0,0,0,0
  726  002014  000000
  727                                  ;ITEM 0
  728  002016  000000  000000  000000          .WORD   0,0,0,0
  729  002024  000000
  730                                  ;ITEM 0
  731  002026  000000  000000  000000          .WORD   0,0,0,0
  732  002034  000000
  733                                  ;ITEM 0
  734  002036  000000  000000  000000          .WORD   0,0,0,0
  735  002044  000000
  736                                  ;ITEM 0
  737  002046  000000  000000  000000          .WORD   0,0,0,0
  738  002054  000000
  739
  740                                  ;ITEM 55
  741  002056  035301  046650  050636          .WORD   EM55,DH55,DT55,DF55
  742  002064  050420
  743                                  ;ITEM 56
  744  002066  035445  046650  050636          .WORD   EM56,DH56,DT56,DF56
  745  002074  050420
  746                                  ;ITEM 57
  747  002076  035612  046650  050636          .WORD   EM57,DH57,DT57,DF57
  748  002104  050420
  749                                  ;ITEM 60
  750  002106  035734  046650  050636          .WORD   EM60,DH60,DT60,DF60
  751  002114  050420
  752                                  ;ITEM 61
  753  002116  036060  046650  050636          .WORD   EM61,DH61,DT61,DF61
  754  002124  050420
```

```
                                                ;ITEM 62
002126  036210  046650  050636                        .WORD   EM62,DH62,DT62,DF62
002134  050420
                                                ;ITEM 63
002136  036336  046725  050650                        .WORD   EM63,DH63,DT63,DF63
002144  050424
                                                ;ITEM 64
002146  036555  047027  050662                        .WORD   EM64,DH64,DT64,DF64
002154  050424
                                                ;ITEM 65
002156  036752  047102  050672                        .WORD   EM65,DH65,DT65,DF65
002164  050424
                                                ;ITEM 66
002166  037335  047204  050704                        .WORD   EM66,DH66,DT66,DF66
002174  050424
                                                ;ITEM 67
002176  037417  047257  050662                        .WORD   EM67,DH67,DT67,DF67
002204  050424
                                                ;ITEM 70
002206  037634  047257  050662                        .WORD   EM70,DH70,DT70,DF70
002214  050424
                                                ;ITEM 71
002216  040112  047257  050662                        .WORD   EM71,DH71,DT71,DF71
002224  050424
                                                ;ITEM 72
002226  040370  047257  050662                        .WORD   EM72,DH72,DT72,DF72
002234  050424
                                                ;ITEM 73
002236  040612  047257  050662                        .WORD   EM73,DH73,DT73,DF73
002244  050424
                                                ;ITEM 74
002246  041076  047257  050662                        .WORD   EM74,DH74,DT74,DF74
002254  050424
                                                ;ITEM 75
002256  041362  047354  050720                        .WORD   EM75,DH75,DT75,DF75
002264  050431
                                                ;ITEM 76
002266  041362  047354  050734                        .WORD   EM76,DH76,DT76,DF76
002274  050431
                                                ;ITEM 77
002276  041521  047451  050750                        .WORD   EM77,DH77,DT77,DF77
002304  050436
                                                ;ITEM 0
002306  000000  000000  000000                        .WORD   0,0,0,0
002314  000000
                                                ;ITEM 0
002316  000000  000000  000000                        .WORD   0,0,0,0
002324  000000
                                                ;ITEM 0
002326  000000  000000  000000                        .WORD   0,0,0,0
002334  000000
                                                ;ITEM 0
002336  000000  000000  000000                        .WORD   0,0,0,0
002344  000000
                                                ;ITEM 0
```

```
  811  002346  000000  000000  000000          .WORD   0,0,0,0
  812  002354  000000
  813                                  ;ITEM 0
  814  002356  000000  000000  000000          .WORD   0,0,0,0
  815  002364  000000
  816                                  ;ITEM 0
  817  002366  000000  000000  000000          .WORD   0,0,0,0
  818  002374  000000
  819                                  ;ITEM 0
  820  002376  000000  000000  000000          .WORD   0,0,0,0
  821  002404  000000
  822                                  ;ITEM 0
  823  002406  000000  000000  000000          .WORD   0,0,0,0
  824  002414  000000
  825                                  ;ITEM 0
  826  002416  000000  000000  000000          .WORD   0,0,0,0
  827  002424  000000
  828                                  ;ITEM 0
  829  002426  000000  000000  000000          .WORD   0,0,0,0
  830  002434  000000
  831
  832                                  ;ITEM 0
  833  002436  000000  000000  000000          .WORD   0,0,0,0
  834  002444  000000
  835                                  ;ITEM 0
  836  002446  000000  000000  000000          .WORD   0,0,0,0
  837  002454  000000
  838                                  ;ITEM 0
  839  002456  000000  000000  000000          .WORD   0,0,0,0
  840  002464  000000
  841                                  ;ITEM 0
  842  002466  000000  000000  000000          .WORD   0,0,0,0
  843  002474  000000
  844                                  ;ITEM 117
  845  002476  041657  047354  050734          .WORD   EM117,DH117,DT117,DF117
  846  002504  050431
  847                                  ;ITEM 120
  848  002506  042006  047475  050776          .WORD   EM120,DH120,DT120,DF120
  849  002514  050450
  850                                  ;ITEM 121
  851  002516  042221  047551  051066          .WORD   EM121,DH121,DT121,DF121
  852  002524  050503
  853                                  ;ITEM 122
  854  002526  042422  047613  051100          .WORD   EM122,DH122,DT122,DF122
  855  002534  050507
  856                                  ;ITEM 123
  857  002536  042552  047675  051100          .WORD   EM123,DH123,DT123,DF123
  858  002544  050507
  859                                  ;ITEM 124
  860  002546  042753  046531  051112          .WORD   EM124,DH124,DT124,DF124
  861  002554  050513
  862                                  ;ITEM 0
  863  002556  000000  000000  000000          .WORD   0,0,0,0
  864  002564  000000
  865                                  ;ITEM 0
  866  002566  000000  000000  000000          .WORD   0,0,0,0
```

```
867  002574  000000
868                                  ;ITEM 127
869  002576  043161  050045  051132          .WORD   EM127,DH127,DT127,DF127
870  002604  050537
871                                  ;ITEM 130
872  002606  043343  050107  051164          .WORD   EM130,DH130,DT130,DF130
873  002614  050523
874
875                                  ;ITEM 131
876  002616  043415  050165  051176          .WORD   EM131,DH131,DT131,DF131
877  002624  050542
878                                  ;ITEM 132
879  002626  045530  047735  051132          .WORD   EM132,DH132,DT132,DF132
880  002634  050523
881                                  ;ITEM 133
882  002636  045667  047772  051142          .WORD   EM133,DH133,DT133,DF133
883  002644  050527
884                                  ;ITEM 134
885  002646  046041  050244  051224          .WORD   EM134,DH134,DT134,DF134
886  002654  050554
887                                  ;ITEM 135
888  002656  046207  047451  051244          .WORD   EM135,DH135,DT135,DF135
889  002664  050563
890                                  ;ITEM 0
891  002666  000000  000000  000000          .WORD   0,0,0,0
892  002674  000000
893                                  ;ITEM 0
894  002676  000000  000000  000000          .WORD   0,0,0,0
895  002704  000000
896                                  ;ITEM 140
897  002706  043642  045446  045516          .WORD   EM140,DH140,DT140,DF140
898  002714  045511
899                                  ;ITEM 141
900  002716  044203  045446  045516          .WORD   EM141,DH141,DT141,DF141
901  002724  045511
902                                  ;ITEM 142
903  002726  044543  045446  045516          .WORD   EM142,DH142,DT142,DF142
904  002734  045511
905                                  ;ITEM 143
906  002736  045105  045446  045516          .WORD   EM143,DH143,DT143,DF143
907  002744  045511
908                                  ;ITEM 0
909  002746  000000  000000  000000          .WORD   0,0,0,0
910  002754  000000
911                                  ;ITEM 0
912  002756  000000  000000  000000          .WORD   0,0,0,0
913  002764  000000
914                                  ;ITEM 0
915  002766  000000  000000  000000          .WORD   0,0,0,0
916  002774  000000
917                                  ;ITEM 0
918  002776  000000  000000  000000          .WORD   0,0,0,0
919  003004  000000
920                                  ;ITEM 150
921  003006  046372  050321  051272          .WORD   EM150,DH150,DT150,DF150
922  003014  050575
```

```
                                        START:  CLR     $TSTNM
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
003016  005037  001102                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
003022  012706  001100                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
003026  005026                                  CMP     #SWR,R6 ;;DONE?
003030  022706  001140                          BNE     .-6             ;;LOOP BACK IF NO
003034  001374                                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
003036  012706  001100                  ;;INITIALIZE A FEW VECTORS
003042  012737  026460  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
003050  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
003056  012737  026736  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
003064  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
003072  012737  030104  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
003100  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
003106  012737  030202  000024                  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
003114  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
003122  013737  026354  026346                  MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
003130  005037  001302                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
003134  005037  001304                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
003140  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
003146  012737  003146  001106                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
003154  012737  003154  001110                  MOV     #.,$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.
003162  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
003166  012737  003222  000004                  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
003174  012737  177570  001140                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003202  012737  177570  001142                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003210  022777  177777  175722                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003216  001012                                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
003220  000403                                  BR      65$             ;;BRANCH IF NO TIMEOUT
003222  012716  003230                  64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
003226  000002                                  RTI
003230  012737  000176  001140          65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
003236  012737  000174  001142                  MOV     #DISPREG,DISPLAY
003244  012637  000004                  66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

                                        .SBTTL  TYPE PROGRAM NAME
                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
003250  005227  177777                          INC     #-1             ;;FIRST TIME?
003254  001043                                  BNE     67$             ;;BRANCH IF NO
003256  022737  026424  000042                  CMP     #$ENDAD,@#42    ;;ACT-11?
003264  001437                                  BEQ     67$             ;;BRANCH IF YES
003266  104401  003274                          TYPE    ,68$            ;;TYPE ASCIZ STRING
003272  000434                                  BR      67$             ;;GET OVER THE ASCIZ
                                        ;;68$:  .ASCIZ  <CRLF>'MAINDEC-11-DEKBC-B  PDP 11/70 CACHE DIAGNOSTIC PART 1'<CRLF>
003364                                  67$:
                                        ;THIS ROUTINE SAVES THE TOP 1500 (DEC) WORDS OF THE FIRST 28K OF
                                        ;MEMORY. THESE LOCATIONS SHOULD CONTAIN EITHER THE MONITOR OR THE
                                        ;LOADER WHICH LOADED THE PROGRAM. NOTE THAT TO RESTORE THIS PART
                                        ;OF CORE, THAT IS TO RESTORE THE LOADER OR MONITOR, ALL THE USER
```

```
                                        ;MUST DO IS TYPE ↑C (CONTROL-C), WHILE THIS PROGRAM IS RUNNING.
                                        ;THIS WILL AUTOMATICALLY RESTORE THE TOP PART OF MEMORY TO ITS STATE
                                        ;BEFORE THIS PROGRAM WAS STARTED! AFTER THE MONITOR (OR LOADER) HAS BEEN
                                        ;RESTORED THIS PROGRAM WILL HALT.
003364  005237  031252          LOOP:   INC     MONF            ;INCREMENT THE FLAG WHICH INDICATES
003370  001013                          BNE     TOP             ;WHETHER OR NOT THE TOP OF MEMORY
                                                                ;IN THE FIRST 28K HAS BEEN SAVED.
003372  013737  000060  031250          MOV     @#TKVEC,MONTTY  ;SAVE THE INITIAL CONTENTS OF THE TTY KEYBOARD
                                                                ;VECTOR.
003400  012700  002734                  MOV     #↑D1500,R0      ;IF NOT THEN SAVE IT.
003404  012701  051310                  MOV     #BOTTOM+4,R1    ;SAVE IT AT THE BOTTOM OF THIS PROGRAM.
003410  012702  160000                  MOV     #160000,R2      ;GET THE ADDRESS OF THE END OF THE MONITOR.
003414  014221                  1$:     MOV     -(R2),(R1)+     ;SAVE 1500 (DEC) LOCATIONS (WORDS)
003416  077002                          SOB     R0,1$
003420  012737  000044  177770  TOP:    MOV     #44,@#177770    ;SET TO SYNC SCOPE (OSCILLOSCOPE)
                                                                ;ON A NOP INSTRUCTION.

003426  012737  031130  000060          MOV     #RESMON,@#TKVEC ;SET UP THE KEYBORD INTERRUPT VECTOR.
003434  012737  000340  000062          MOV     #340,@#↑KVEC+2
003442  005077  17550C                  CLR     @$TKB           ;MAKE SURE THE BUFFER IS CLEAR
003446  152777  000100  175470          BISB    #BIT6,@$TKS     ;TURN ON INTERRUPT ENABLE FOR THE KEYBOARD.

003454  012737  030474  000004          MOV     #CPSPUR,@#ERRVEC        ;SET UP FOR UNEXPECTED ERRORS.
003462  012737  030522  000114          MOV     #SPUR,@#CACHVEC


                                        ;;**********************************************************************
                                        ;*TEST 1         CACHE REGISTERS RESPONSE TEST
                                        ;*
                                        ;*REFERENCE EACH CACHE REGISTER MAKING SURE SUCH
                                        ;*REFERENCES DO NOT TIME OUT.
                                        ;*
                                        ;;**********************************************************************
003470  000004                  TST1:   SCOPE
003472  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000001                  JA=$TN-1
                                                                ;SET THE SKAD REGISTER
003500  012737  004044  030646          MOV     #TST2,SKAD      ;IN CASE THE TEST ABORTS.

003506  113737  001102  001232          MOVB    $TSTNM,$TMPO
003514  012737  030522  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.
003522  012701  031056                  MOV     #LOAFLG,R1      ;CLEAR THE REGISTER FLAGS
003526  012700  000014                  MOV     #14,R0
003532  005021                  64$:    CLR     (R1)+
003534  077002                          SOB     R0,64$
003536  013737  000004  003614          MOV     @#ERRVEC,JATMP  ;SAVE THE OLD CONTENTS OF VECTOR ERRVEC.
003544  012737  003616  000004          MOV     #JAERR,@#ERRVEC ;SET UP THE TIME OUT
                                                                ;VECTOR
003552  012700  177740                  MOV     #LOADRS,R0
003556  012737  003564  001110          MOV     #JA1,$LPERR

003564  000240                  JA1:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
003566  005710                          TST     (R0)            ;REFERENCE EACH CACHE REGISTER
                                                                ;MAKING SURE EACH DOESN'T TIME OUT.

003570  062700  000002          JA2:    ADD     #2,R0
```

```
1035  003574  020027  177752              CMP     R0,#HITMIS
1036  003600  101771                      BLOS    JA1
1037
1038  003602  013737  003614  000004 JA3:  MOV     JATMP,@#ERRVEC  ;RESET THE CPU TRAP VECTOR.
1039  003610  000137  004040              JMP     JADONE
1040
1041  003614  000000         JATMP:       .WORD   0               ;SAVE THE OLD CONTENTS OF
1042                                                               ;VECTOR ERRVEC HERE.
1043
1044  003616  032737  000020  177766 JAERR:  BIT     #20,@#CPUERR
1045  003624  001005                      BNE     JAERR1          ;MAKE SURE THE ERROR
1046  003626  013737  003614  000004 JAERR0: MOV     JATMP,@#ERRVEC  ;IF NOT RESET VECTOR ERRVEC AND GO TO
1047  003634  000177  174144              JMP     @ERRVEC         ;THE ROUTINE WHICH HANDLES CPU ERRORS.
1048  003640  021627  003570         JAERR1: CMP     (SP),#JA2       ;OTHERWISE REPORT THE FACT THAT A CACHE
1049  003644  001370                      BNE     JAERR0          ;REGISTER REFERENCE TIMED OUT!
1050  003646  012637  001234              MOV     (SP)+,$TMP1
1051  003652  005726                      TST     (SP)+
1052  003654  010037  001240              MOV     R0,$TMP3
1053  003660  012737  000077  001242      MOV     #77,$TMP4
1054  003666  020027  17774C              CMP     R0,#LOADRS
1055  003672  001005                      BNE     JAERR2
1056  003674  012737  177777  031056      MOV     #-1,LOAFLG
1057  003702  104055         1$:          ERROR   55
1058  003704  000451                      BR      JAERR9
1059
1060  003706  020027  177742         JAERR2: CMP     R0,#HIADRS
1061  003712  001005                      BNE     JAERR3
1062  003714  012737  177777  031060      MOV     #-1,HIAFLG
1063  003722  104056         1$:          ERROR   56
1064  003724  000441                      BR      JAERR9
1065
1066  003726  020027  177744         JAERR3: CMP     R0,#MEMERR
1067  003732  001005                      BNE     JAERR4
1068  003734  012737  177777  031062      MOV     #-1,MMRFLG
1069  003742  104057         1$:          ERROR   57
1070  003744  000431                      BR      JAERR9
1071
1072  003746  020027  177746         JAERR4: CMP     R0,#CONTRL
1073  003752  001005                      BNE     JAERR5
1074  003754  012737  177777  031064      MOV     #-1,CONFLG
1075  003762  104060         1$:          ERROR   60
1076  003764  000421                      BR      JAERR9
1077
1078  003766  020027  177750         JAERR5: CMP     R0,#MAINT
1079  003772  001005                      BNE     JAERR6
1080  003774  012737  177777  031066      MOV     #-1,MANFLG
1081  004002  104061         1$:          ERROR   61
1082  004004  000411                      BR      JAERR9
1083
1084  004006  020027  177752         JAERR6: CMP     R0,#HITMIS
1085  004012  001005                      BNE     JAERR7
1086  004014  012737  177777  031070      MOV     #-1,HIMFLG
1087  004022  104062         1$:          ERROR   62
1088  004024  000401                      BR      JAERR9
1089
1090  004026  000000         JAERR7:      HALT                    ;???
```

```
1091
1092  004030  005037  177766          JAERR9: CLR     @#CPUERR
1093  004034  000137  003570                  JMP     JA2
1094
1095  004040  005037  177766          JADONE: CLR     @#CPUERR        ;DONE!
1096
1097                                  ;;*****************************************************************
1098                                  ;*TEST 2        CACHE REGISTERS DATA PATH, READ ZEROES TEST
1099                                  ;*
1100                                  ;*THIS TEST CHECKS THE ABILITY OF THE CACHE REGISTER
1101                                  ;*DATA PATHS TO PASS 0'S BY FIRST WRITING THEN READING
1102                                  ;*0'S AT THE CONTROL AND MAINTENANCE REGISTERS.
1103                                  ;*
1104                                  ;;*****************************************************************
1105  004044  000004                  TST2:   SCOPE
1106          000002                  JB=$TN-1
1107                                                                  ;SET THE SKAD REGISTER
1108  004046  012737  004200  030646          MOV     #TST3,SKAD      ;IN CASE THE TEST ABORTS.
1109
1110  004054  113737  001102  001232          MOVB    $TSTNM,$TMPO
1111  004062  012737  030522  000114          MOV     #SPUR,@#CACHVEC
1112
1113  004070  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1114  004072  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
1115  004074  012737  004102  001110          MOV     #JB1,$LPERR
1116  004102  005037  177746          JB1:    CLR     @#CONTRL        ;WRITE ZEROES
1117  004106  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
1118  004110  013700  177746                  MOV     @#CONTRL,R0     ;READ.ZEROES
1119  004114  005700                          TST     R0
1120  004116  001430                          BEQ     JBDONE
1121  004120  005037  177750          JB2:    CLR     @#MAINT
1122  004124  013701  177750                  MOV     @#MAINT,R1
1123  004130  005701                          TST     R1
1124  004132  001414                          BEQ     JBERR2
1125
1126  004134                          JBERR1:                         ;BOTH READ ZEROES FAILED.
1127  004134  010037  001236                  MOV     R0,$TMP2
1128  004140  010137  001240                  MOV     R1,$TMP3
1129  004144  104063                  1$:     ERROR   63
1130  004146  012737  177777  031064          MOV     #-1,CONFLG      ;SIGNAL BAD REGISTERS
1131  004154  012737  177777  031066          MOV     #-1,MANFLG
1132  004162  000406                          BR      JBDONE
1133
1134  004164                          JBERR2:                         ;ONLY THE READ OF THE
1135  004164  010037  001236                  MOV     R0,$TMP2        ;CONTROL REGISTER FAILED.
1136  004170  104064                  1$:     ERROR   64
1137  004172  012737  177777  031064          MOV     #-1,CONFLG
1138
1139  004200                          JBDONE:                         ;DONE!!!
1140
1141                                  ;;*****************************************************************
1142                                  ;*TEST 3        CACHE REGISTERS DATA PATH, READ ONES TEST
1143                                  ;*
1144                                  ;*THIS TEST PERFORMS A READ OF BOTH THE HIGH ORDER AND
1145                                  ;*LOW ORDER ERROR ADDRESS REGISTER. THIS IS DONE TO MAKE
1146                                  ;*SURE THAT THE REGISTERS' DATA PATHS CAN PASS ONES. NOTE THAT
```

```
                                        ;*THE LOW ORDER ADDRESS REGISTER SHOULD CONTAIN A
                                        ;*177740 AND THE HIGH ORDER REGISTER SHOULD CONTAIN
                                        ;*000003. THIS LEAVES THE DATA PATH LINE'S BITS 2,3 AND 4
                                        ;*UNTESTED FOR THEIR AVAILITY TO PASS ONES. THIS WILL
                                        ;*BE CHECKED IN THE COUNT PATTERN TST4.
                                        ;*
                                        ;;*************************************************************************
004200  000004                    TST3:   SCOPE
004202  012737  000040  001302            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000003                  JC=$TN-1
                                                                    ;SET THE SKAD REGISTER
004210  012737  004342  030646            MOV     #TST4,SKAD      ;IN CASE THE TEST ABORTS.

004216  113737  001102  001232            MOVB    $TSTNM,$TMPO

004224  104414                          SKPBAD          ;IF THE ERROR ADDRESS REG IS BAD SKIP THIS TEST.
004226  104415                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
004230  012737  177777  177744          MOV     #-1,@#MEMERR    ;MAKE SURE THE ERROR REGISTERS ARE UNLOCKED
004236  012737  004244  001110          MOV     #JC1,$LPERR

004244  000240                  JC1:    NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
004246  013700  177740                  MOV     @#LOADRS,R0
004252  013701  177742                  MOV     @#HIADRS,R1     ;READ THE REGISTERS.
004256  022700  177740                  CMP     #177740,R0
004262  001003                          BNE     JCERR1
004264  022701  000003          JC2:    CMP     #3,R1
004270  001424                          BEQ     JCDONE

004272  012737  004310  001234  JCERR1: MOV     #1$,$TMP1       ;BAD DATA WAS READ FROM THEM!!
004300  010037  001236                  MOV     R0,$TMP2
004304  010137  001240                  MOV     R1,$TMP3
004310  104065                  1$:     ERROR   65
004312  022700  000003                  CMP     #3,R0
004316  001403                          BEQ     2$
004320  012737  177777  031056          MOV     #-1,LOAFLG
004326  022700  177740          2$:     CMP     #177740,R0
004332  001403                          BEQ     JCDONE
004334  012737  177777  031060          MOV     #-1,HIAFLG

004342                          JCDONE:                         ;DONE!


                                        ;;*************************************************************************
                                        ;*TEST 4         CACHE CONTROL REGISTER COUNT PATTERN TEST
                                        ;*
                                        ;*THIS TEST RUNS A COUNT PATTERN THROUGH THE CACHE CONTROL
                                        ;*REGISTER FOR THE PURPOSE OF CHECKING OUT THE
                                        ;*DATA RELIABILITY OF BOTH THE REGISTER BITS AND THE
                                        ;*DATA PATHS LINES.
                                        ;*
                                        ;;*************************************************************************
004342  000004                    TST4:   SCOPE
004344  012737  000004  001302            MOV     #4,$TIMES       ;;DO 4 ITERATIONS

        000004                  JD=$TN-1
```

```
1203                                                                    ;SET THE SKAD REGISTER
1204  004352  012737  004460  030646          MOV     #TST5,SKAD        ;IN CASE THE TEST ABORTS.
1205
1206  004360  113737  001102  001232          MOVB    $TSTNM,$TMP0
1207
1208
1209  004366  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1210
1211  004370  012700  177746                  MOV     #CONTRL,R0
1212  004374  005002                          CLR     R2
1213  004376  012737  004404  001110          MOV     #JD1,$LPERR
1214  004404  000240                  JD1:    NOP                       ;FOR SCOPING WITH AN OSCILLOSCOPE!
1215  004406  010210                          MOV     R2,(R0)           ;WRITE THE REGISTER.
1216  004410  011001                          MOV     (R0),R1           ;READ BACK THE REGISTER AND MAKE SURE
1217  004412  010203                          MOV     R2,R3             ;THE DATA IS CORRECT.
1218  004414  042703  177700                  BIC     #177700,R3
1219  004420  020301                          CMP     R3,R1
1220  004422  001003                          BNE     JDERR1
1221  004424  077211                  JD2:    SOB     R2,JD1
1222  004426  005010                          CLR     (R0)
1223  004430  000413                          BR      JDDONE
1224  004432                          JDERR1:                           ;REPORT THE ERROR!
1225  004432  010237  001236                  MOV     R2,$TMP2
1226  004436  010137  001240                  MOV     R1,$TMP3
1227  004442  010337  001242                  MOV     R3,$TMP4
1228  004446  104066                  1$:     ERROR   66
1229  004450  012737  177777  031064          MOV     #-1,CONFLG
1230  004456  000762                          BR      JD2
1231  004460                          JDDONE:
1232
1233
1234                                    ;;*********************************************************************
1235                                    ;*TEST 5        CACHE HIT/MISS AND CONTROL REGISTER SIMPLE MISSES TEST
1236                                    ;*
1237                                    ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
1238                                    ;*CONTRL REGISTER'S ABILITY TO FORCE MISSES. ZEROES ARE
1239                                    ;*FLOATED THROUGH THE HIT/MISS REGISTER.
1240                                    ;*
1241                                    ;;*********************************************************************
1242  004460  000004                  TST5:   SCOPE
1243  004462  012737  000040  001302          MOV     #40,$TIMES        ;;DO 40 ITERATIONS
1244          000005                  KB=$TN-1
1245                                                                    ;SET THE SKAD REGISTER
1246  004470  012737  005012  030646          MOV     #TST6,SKAD        ;IN CASE THE TEST ABORTS.
1247
1248  004476  113737  001102  001232          MOVB    $TSTNM,$TMP0
1249
1250
1251  004504  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1252  004506  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1253  004510  005037  004702                  CLR     KBFLG
1254  004514  012737  000014  177746  KB1:    MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
1255  004522  012737  004514  001110          MOV     #KB1,$LPERR
1256
1257  004530  012700  004540                  MOV     #KB2,R0
1258  004534  012701  000020                  MOV     #20,R1
```

```
1259  004540  005720                    KB2:    TST     (R0)+
1260  004542  077102                            SOB     R1,KB2
1261  004544  000240                            NOP                     ;GET SIX FORCED MISSES.
1262  004546  000240                            NOP
1263  004550  000240                            NOP
1264  004552  000240                            NOP
1265  004554  013702  177752                    MOV     @#HITMIS,R2     ;SHOULD HAVE REGISTERED
1266  004560  001051                            BNE     KBERR1          ;SIX MISSES.
1267
1268  004562  012737  004562  001110  KB3:     MOV     #KB3,$LPERR
1269  004570  012737  000054  177746            MOV     #S1M0M1,@#CONTRL         ;SELECT GROUP ONE, MISS GROUP
1270  004576  012700  004606                    MOV     #KB4,R0         ;ZERO AND GROUP ONE.
1271  004602  012701  000020                    MOV     #20,R1
1272  004606  005720                    KB4:    TST     (R0)+
1273  004610  077102                            SOB     R1,KB4
1274  004612  000240                            NOP
1275  004614  000240                            NOP
1276  004616  000240                            NOP
1277  004620  000240                            NOP
1278  004622  013702  177752                    MOV     @#HITMIS,R2     ;SHOULD HAVE SIX MISSES.
1279  004626  001035                            BNE     KBERR2
1280
1281  004630  012737  004630  001110  KB5:     MOV     #KB5,$LPERR
1282  004636  012737  000034  177746            MOV     #S0M0M1,@#CONTRL         ;SELECT GROUP 0, MISS GROUP 0
1283  004644  012700  004654                    MOV     #KB6,R0         ;AND GROUP 1.
1284  004650  012701  000020                    MOV     #20,R1
1285  004654  005720                    KB6:    TST     (R0)+
1286  004656  077102                            SOB     R1,KB6
1287  004660  000240                            NOP
1288  004662  000240                            NOP
1289  004664  000240                            NOP
1290  004666  000240                            NOP
1291  004670  013702  177752                    MOV     @#HITMIS,R2     ;SHOULD HAVE SIX MISSES.
1292  004674  001021                            BNE     KBERR3
1293  004676  000137  004754                    JMP     KBDONE
1294
1295
1296  004702  000000            KBFLG:  .WORD   0               ;ERROR FLAG.
1297
1298  004704                    KBERR1:                         ;GOT HITS WHILE FORCING
1299  004704  010237  001236                    MOV     R2,$TMP2        ;MISSES TO BOTH GROUPS.
1300  004710  104072            1$:     ERROR   72
1301  004712  052737  000001  004702            BIS     #BIT0,KBFLG
1302  004720  000720                            BR      KB3
1303  004722                    KBERR2:                         ;GO HITS WHILE FORCING
1304  004722  010237  001236                    MOV     R2,$TMP2        ;MISSES TO BOTH GROUPS
1305  004726  104073            1$:     ERROR   73              ;AND SELECTING GROUP 1
1306  004730  052737  000002  004702            BIS     #BIT1,KBFLG
1307  004736  000734                            BR      KB5
1308  004740                    KBERR3:                         ;GO HITS WHILE FORCING
1309  004740  010237  001236                    MOV     R2,$TMP2        ;MISSES TO BOTH GROUPS
1310  004744  104074            1$:     ERROR   74              ;AND SELECTING GROUP 0.
1311  004746  052737  000004  004702            BIS     #BIT2,KBFLG
1312
1313  004754  005037  177746  KBDONE: CLR     @#CONTRL
1314  004760  022737  000007  004702            CMP     #7,KBFLG        ;IF THE TEST DETECTED
```

```
1315  004766  001003                          BNE     KBD2            ;HITS FOR ALL OF THE
1316  004770  012737  177777  031104          MOV     #-1,HIMFL2      ;THREE CONDITION USED IN
1317                                                                    ;THE CONTROL REGISTER
1318                                                                    ;SIGNAL A BAD HIT/MISS
1319                                                                    ;REGISTER.
1320  004776  005737  004702          KBD2:   TST     KBFLG           ;IF LESS THEN THREE (BUT
1321  005002  001403                          BEQ     KBD3            ;MORE THAN ZERO) CONTRL
1322  005004  012737  177777  031100          MOV     #-1,CONFL2      ;PATTERNS FAILED SIGNAL
1323                                                                    ;A BAD CONTROL REGISTER.
1324  005012                            KBD3:                         ;DONE!
1325
1326                                    ;;*****************************************************************
1327                                    ;*TEST 6        CACHE HIT/MISS AND CONTROL REGISTER SIMPLE HIT TEST
1328                                    ;*
1329                                    ;*THIS IS A TEST OF THE HIT/MISS REGISTER AND THE
1330                                    ;*THE FORCE MISS BITS OF THE CONTROL REGISTER.
1331                                    ;*WHAT IS DONE IS TO SEE IF ANY HITS AT ALL ARE
1332                                    ;*POSSIBLE WITH THE CONTROL REGISTER CLEARED.  THEN THE
1333                                    ;*SAME IS DONE WITH EACH GROUP DISABLE ONE AT A TIME.
1334                                    ;*BY DISABLED IS MEANT THAT THE FORCE MISS BIT IS SET
1335                                    ;*IN THE CONTROL REGISTER FOR THE DISABLED GROUP AND THE
1336                                    ;*FORCE SELECT BIT IS SET FOR THE OTHER GROUP.
1337                                    ;*
1338                                    ;;*****************************************************************
1339  005012  000004            TST6:   SCOPE
1340  005014  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1341           000006                 KA=$TN-1
1342                                                                    ;SET THE SKAD REGISTER
1343  005022  012737  005362  030646          MOV     #TST7,SKAD      ;IN CASE THE TEST ABORTS.
1344
1345  005030  113737  001102  001232          MOVB    $TSTNM,$TMPO
1346
1347
1348  005036  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1349  005040  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1350  005042  005037  005246                  CLR     KAFLG
1351  005046  005037  177746          KA1:    CLR     @#CONTRL        ;BOTH GROUPS ENABLED.
1352  005052  012737  005046  001110          MOV     #KA1,$LPERR
1353  005060  012700  005070                  MOV     #KA2,R0
1354  005064  012701  000020                  MOV     #20,R1
1355
1356  005070  005720            KA2:    TST     (R0)+           ;SET UP HITS IN BOTH
1357  005072  077102                    SOB     R1,KA2          ;GROUPS
1358  005074  000240                    NOP
1359  005076  000240                    NOP
1360  005100  000240                    NOP
1361  005102  000240                    NOP
1362  005104  013702  177752                  MOV     @#HITMIS,R2     ;SHOULD HAVE ALL HITS.
1363  005110  022702  000077                  CMP     #77,R2
1364  005114  001055                          BNE     KAERR1
1365
1366  005116  012737  005116  001110  KA3:    MOV     #KA3,$LPERR
1367  005124  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;DISABLE GROUP ZERO.
1368  005132  012700  005142                  MOV     #KA4,R0
1369  005136  012701  000020                  MOV     #20,R1
1370  005142  005720            KA4:    TST     (R0)+           ;SET UP HITS IN GROUP 1
```

```
1371  005144  077102                          SOB     R1,KA4
1372  005146  000240                          NOP
1373  005150  000240                          NOP
1374  005152  000240                          NOP
1375  005154  000240                          NOP
1376  005156  013702  177752                  MOV     @#HITMIS,R2     ;SHOULD HAVE ALL HITS.
1377  005162  022702  000077                  CMP     #77,R2
1378  005166  001037                          BNE     KAERR2
1379  005170  012737  005170  001110  KA5:    MOV     #KA5,$LPERR
1380  005176  012737  000030  177746          MOV     #SOMI,@#CONTRL  ;DISABLE GROUP ONE.
1381  005204  012700  005214                  MOV     #KA6,R0
1382  005210  012701  000020                  MOV     #20,R1
1383  005214  005720                  KA6:    TST     (R0)+           ;SET UP HITS IN GROUP ZERO.
1384  005216  077102                          SOB     R1,KA6
1385  005220  000240                          NOP
1386  005222  000240                          NOP
1387  005224  000240                          NOP
1388  005226  000240                          NOP
1389  005230  013702  177752                  MOV     @#HITMIS,R2     ;SHOULD HAVE SIX HITS.
1390  005234  022702  000077                  CMP     #77,R2
1391  005240  001021                          BNE     KAERR3
1392  005242  000137  005320                  JMP     KADONE
1393
1394  005246  000000                  KAFLG:  .WORD   0               ;ERROR FLAG.
1395
1396  005250                          KAERR1:                         ;FAILED TO GET HITS
1397  005250  010237  001236                  MOV     R2,$TMP2        ;WITH THE CONTROL
1398  005254  104067                  1$:     ERROR   67              ;REGISTER CLEAR!
1399  005256  052737  000001  005246          BIS     #BIT0,KAFLG
1400  005264  000714                          BR      KA3
1401  005266                          KAERR2:                         ;FAILED TO GET HITS
1402  005266  010237  001236                  MOV     R2,$TMP2        ;WITH THE CONTROL REGISTER
1403  005272  104070                  1$:     ERROR   70              ;SET TO FORCE SELECT GROUP
1404  005274  052737  000002  005246          BIS     #BIT1,KAFLG     ;ONE FORCE MISS GROUP ZERO.
1405  005302  000732                          BR      KA5
1406  005304                          KAERR3:                         ;FAILED TO GET HITS
1407  005304  010237  001236                  MOV     R2,$TMP2        ;WITH THE CONTROL REGISER
1408  005310  104071                  1$:     ERROR   71              ;SET TO FORCE SELECT GROUP
1409  005312  052737  000004  005246          BIS     #BIT2,KAFLG     ;ZERO AND FORCE MISS GROUP ONE.
1410  005320  005037  177746          KADONE: CLR     @#CONTRL
1411  005324  022737  000007  005246          CMP     #7,KAFLG        ;IF THE TEST FAILED FOR ALL
1412  005332  001004                          BNE     KAD2            ;THREE CONDITIONS OF THE
1413  005334  012737  177777  031070          MOV     #-1,HIMFLG      ;CONTROL REGISTER SIGNAL
1414  005342  000407                          BR      KAD3            ;A BAD HIT/MISS REGISTER.
1415
1416  005344  032737  000006  005246  KAD2:   BIT     #6,KAFLG        ;IF THE TEST FAILED ONLY WHEN
1417  005352  001403                          BEQ     KAD3            ;THE CONTROL REGISTER WAS SET
1418  005354  012737  177777  031100          MOV     #-1,CONFL2      ;SIGNAL A BAD CONTROL REGISTER.
1419  005362                          KAD3:                           ;DONE!!
1420
1421
1422                                  ;;*****************************************************************
1423                                  ;*TEST 7        CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 0 TEST
1424                                  ;*
1425                                  ;*THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS
1426                                  ;*OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS
```

```
                                        ;*MADE A HIT IN GROUP ONE; THEN ANOTHER ADDRESS, WHOSE
                                        ;*HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS
                                        ;*IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING
                                        ;*SELECTION OF GROUP ZERO; THEN SEE IF THE FIRST ADDRESS
                                        ;*IS STILL A HIT IN GROUP ONE; FINALLY TURN ON THE FORCE
                                        ;*MISS GROUP ZERO BIT AND SEE IF THE SECOND ADDRESS'
                                        ;*HIT IN GROUP ZERO CAN BE FORCED TO A MISS.
                                        ;*
                                        ;;*****************************************************************
005362  000004                TST7:   SCOPE
005364  012737  000040  001302        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000007                  KD=$TN-1
                                                                ;SET THE SKAD REGISTER
005372  012737  005712  030646        MOV     #TST10,SKAD     ;IN CASE THE TEST ABORTS.

005400  113737  001102  001232        MOVB    $TSTNM,$TMP0
005406  012737  030522  000114        MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS.

005414  104416                        SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
005416  104420                        SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.

005420  012700  005710        K1D:    MOV     #KTMP2D,R0      ;DETERMINE THE TEST LOCATIONS.
005424  042700  176003                BIC     #176003,R0
005430  010001                        MOV     R0,R1
005432  062701  140000                ADD     #TESTR1,R1
005436  010137  001252                MOV     R1,$TMP10
005442  005037  001254                CLR     $TMP11
005446  010002                        MOV     R0,R2
005450  062702  142000                ADD     #TESTR2,R2
005454  010237  001256                MOV     R2,$TMP12
005460  005037  001260                CLR     $TMP13

005464  012737  000044  177746  K2D:  MOV     #SIMD,@#CONTRL  ;MAKE (R1) A HIT IN
005472  005711                        TST     (R1)            ;GROUP GRM.
005474  005711                        TST     (R1)
005476  032737  000010  177752        BIT     #10,@#HITMIS
005504  001007                        BNE     K3D

                                                                ;REPORT ERROR, UNABLE
005506  012737  000001  001236        MOV     #1,$TMP2        ;GET A HIT IN GROUP GRM.
005514  012737  000044  001240        MOV     #SIMD,$TMP3
005522  104075                1$:     ERROR   75

005524  012703  000030        K3D:    MOV     #SOM1,R3
005530  042703  000017                BIC     #17,R3
005534  010337  177746                MOV     R3,@#CONTRL     ;FORCE SELECT GROUP GRS.
005540  005712                        TST     (R2)            ;MAKE (R2) A HIT IN GROUP
005542  005712                        TST     (R2)            ;GRS.
005544  032737  000010  177752        BIT     #10,@#HITMIS
005552  001006                        BNE     K4D

                                                                ;IF NOT, ERROR UNABLE TO
                                                                ;GET A HIT IN GROUP 0
005554  010337  001240                MOV     R3,$TMP3
005560  104076                1$:     ERROR   76
005562  012737  177777  031100        MOV     #-1,CONFL2
```

```
1483  005570  005037  177746          K4D:    CLR     @#CONTRL        ;NOW MAKE SURE (R1) IS
1484  005574  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
1485  005576  005711                          TST     (R1)            ;STILL A HIT IN GROUP
1486  005600  032737  000010  177752          BIT     #10,@#HITMIS    ;1, THAT IS MAKE SURE
1487  005606  001010                          BNE     K5D             ;GROUP 1 WASN'T WRITTEN
1488                                                                  ;WHILE FORCE SELECTING
1489                                                                  ;GROUP GRS.
1490  005610  012737  000001  001236          MOV     #1,$TMP2
1491  005616  012737  000000  001240          MOV     #0,$TMP3
1492  005624  104077                  1$:     ERROR   77
1493  005626  000424                          BR      K6D
1494  005630  012703  000044          K5D:    MOV     #S1M0,R3        ;NOW SEE IF YOU CAN
1495  005634  042703  000063                  BIC     #63,R3          ;GET A MISS AT (R2)
1496  005640  010337  177746                  MOV     R3,@#CONTRL     ;BY FORCING MISSES
1497  005644  005712                          TST     (R2)            ;TO GRS.
1498  005646  032737  000010  177752          BIT     #10,@#HITMIS
1499  005654  001411                          BEQ     K6D             ;SHOULD BE A MISS,
1500                                                                  ;OTHERWISE ERROR!
1501  005656  012737  000000  001236          MOV     #0,$TMP2
1502  005664  010337  001240                  MOV     R3,$TMP3
1503  005670  104117                  1$:     ERROR   117
1504  005672  012737  177777  031100          MOV     #-1,CONFL2
1505
1506  005700  005037  177746          K6D:    CLR     @#CONTRL
1507  005704  000402                          BR      K7D
1508
1509  005706  000000                  KTMP1D:.WORD    0
1510  005710  000000                  KTMP2D:.WORD    0
1511
1512  005712                          K7D:                            ;DONE!
1513
1514
1515                                  ;;**************************************************************
1516                                  ;*TEST 10       CACHE CONTROL REGISTER, FORCE SELECT-FORCE MISS, GROUP 1 TEST
1517                                  ;*
1518                                  ;*THIS IS A TEST OF THE CONTROL REGISTER FUNCTIONS
1519                                  ;*OF FORCE MISS AND FORCE SELECTION. AN ADDRESS IS
1520                                  ;*MADE A HIT IN GROUP ZERO; THEN ANOTHER ADDRESS, WHOSE
1521                                  ;*HIT WOULD BE MUTUALLY EXCLUSIVE WITH THE FIRST ADDRESS
1522                                  ;*IN ONLY ONE GROUP, IS MADE A HIT WHILE FORCING
1523                                  ;*SELECTION OF GROUP ONE; THEN SEE IF THE FIRST ADDRESS
1524                                  ;*IS STILL A HIT IN GROUP ZERO; FINALLY TURN ON THE FORCE
1525                                  ;*MISS GROUP ONE BIT AND SEE IF THE SECOND ADDRESS'
1526                                  ;*HIT IN GROUP ONE CAN BE FORCED TO A MISS.
1527                                  ;*
1528                                  ;;**************************************************************
1529  005712  000004                  TST10:  SCOPE
1530  005714  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1531          000010                  KE=$TN-1
1532                                                                  ;SET THE SKAD REGISTER
1533  005722  012737  006242  030646          MOV     #TST11,SKAD     ;IN CASE THE TEST ABORTS.
1534
1535  005730  113737  001102  001232          MOVB    $TSTNM,$TMP0
1536  005736  012737  030522  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO ERRORS.
1537
1538  005744  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
```

```
1539  005746  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1540
1541  005750  012700  006240          K1E:    MOV     #KTMP2E,R0      ;DETERMINE THE TEST LOCATIONS.
1542  005754  042700  176003                  BIC     #176003,R0
1543  005760  010001                          MOV     R0,R1
1544  005762  062701  140000                  ADD     #TESTR1,R1
1545  005766  010137  001252                  MOV     R1,$TMP10
1546  005772  005037  001254                  CLR     $TMP11
1547  005776  010002                          MOV     R0,R2
1548  006000  062702  142000                  ADD     #TESTR2,R2
1549  006004  010237  001256                  MOV     R2,$TMP12
1550  006010  005037  001260                  CLR     $TMP13
1551
1552  006014  012737  000030  177746  K2E:    MOV     #SOM1,@#CONTRL  ;MAKE (R1) A HIT IN
1553  006022  005711                          TST     (R1)            ;GROUP GRM.
1554  006024  005711                          TST     (R1)
1555  006026  032737  000010  177752          BIT     #10,@#HITMIS
1556  006034  001007                          BNE     K3E
1557
1558                                                                  ;REPORT ERROR, UNABLE
1559  006036  012737  000000  001236          MOV     #0,$TMP2        ;GET A HIT IN GROUP GRM.
1560  006044  012737  000030  001240          MOV     #SOM1,$TMP3
1561  006052  104075                  1$:     ERROR   75
1562
1563  006054  012703  000044          K3E:    MOV     #S1M0,R3
1564  006060  042703  000017                  BIC     #17,R3
1565  006064  010337  177746                  MOV     R3,@#CONTRL     ;FORCE SELECT GROUP GRS.
1566  006070  005712                          TST     (R2)            ;MAKE (R2) A HIT IN GROUP
1567  006072  005712                          TST     (R2)            ;GRS.
1568  006074  032737  000010  177752          BIT     #10,@#HITMIS
1569  006102  001006                          BNE     K4E
1570                                                                  ;IF NOT, ERROR UNABLE TO
1571                                                                  ;GET A HIT IN GROUP 1
1572  006104  010337  001240                  MOV     R3,$TMP3
1573  006110  104076                  1$:     ERROR   76
1574  006112  012737  177777  031100          MOV     #-1,CONFL2
1575
1576  006120  005037  177746          K4E:    CLR     @#CONTRL        ;NOW MAKE SURE (R1) IS
1577  006124  000240                          NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
1578  006126  005711                          TST     (R1)            ;STILL A HIT IN GROUP
1579  006130  032737  000010  177752          BIT     #10,@#HITMIS    ;0, THAT IS MAKE SURE
1580  006136  001010                          BNE     K5E             ;GROUP 0 WASN'T WRITTEN
1581                                                                  ;WHILE FORCE SELECTING
1582                                                                  ;GROUP GRS.
1583  006140  012737  000000  001236          MOV     #0,$TMP2
1584  006146  012737  000001  001240          MOV     #1,$TMP3
1585  006154  104077                  1$:     ERROR   77
1586  006156  000424                          BR      K6E
1587  006160  012703  000030          K5E:    MOV     #SOM1,R3        ;NOW SEE IF YOU CAN
1588  006164  042703  000063                  BIC     #63,R3          ;GET A MISS AT (R2)
1589  006170  010337  177746                  MOV     R3,@#CONTRL     ;BY FORCING MISSES
1590  006174  005712                          TST     (R2)            ;TO GRS.
1591  006176  032737  000010  177752          BIT     #10,@#HITMIS
1592  006204  001411                          BEQ     K6E             ;SHOULD BE A MISS,
1593                                                                  ;OTHERWISE ERROR!
1594  006206  012737  000001  001236          MOV     #1,$TMP2
```

```
1595  006214  010337  001240                  MOV     R3,$TMP3
1596  006220  104117                  1$:     ERROR   117
1597  006222  012737  177777  031100          MOV     #-1,CONFL2
1598
1599  006230  005037  177746          K6E:    CLR     @#CONTRL
1600  006234  000402                          BR      K7E
1601
1602  006236  000000                  KTMP1E:.WORD    0
1603  006240  000000                  KTMP2E:.WORD    0
1604
1605  006242                          K7E:                            ;DONE!
1606
1607
1608                                  ;;**************************************************************
1609                                  ;*TEST 11       CACHE HIT/MISS REGISTER PATTERNS TEST
1610                                  ;*
1611                                  ;*THIS IS A TEST OF THE HIT/MISS REGISTER WHICH
1612                                  ;*FLOATS DIFFERENT PATTERNS OF HITS AND MISSES
1613                                  ;*THROUGH THAT REGISTER. THIS IS DONE FIRST WITH
1614                                  ;*BOTH GROUPS ENABLE; THEN WITH GROUP ZERO DISABLED
1615                                  ;*THAT IS FORCING SELECTION OF GROUP ONE AND FORCING
1616                                  ;*MISSES TO GROUP ZERO; FINALLY WITH GROUP ONE
1617                                  ;*DISABLED.
1618                                  ;*
1619                                  ;;**************************************************************
1620  006242  000004                  TST11:  SCOPE
1621  006244  012737  000020  001302          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
1622          000011                  KC=$TN-1
1623                                                                  ;SET THE SKAD REGISTER
1624  006252  012737  007054  030646          MOV     #TST12,SKAD     ;IN CASE THE TEST ABORTS.
1625
1626  006260  113737  001102  001232          MOVB    $TSTNM,$TMP0
1627  006266  012737  030522  000114          MOV     #SPUR,@#CACHVEC
1628
1629  006274  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
1630  006276  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
1631  006300  005037  006736                  CLR     KCCON           ;TEST THE BOTH GROUPS
1632  006304  012737  000002  006740          MOV     #2,KCFLG1       ;ENABLED CONDITION FIRST.
1633  006312  012737  006326  001110  KC0:    MOV     #KC1,$LPERR
1634  006320  012737  006744  006742          MOV     #KCTBL,KCPTR    ;KCPTR IS A POINTER TO
1635                                                                  ;THE TABLE OF 12-BIT PATTERNS
1636                                                                  ;WHICH WILL BE FLOATED
1637                                                                  ;THROUGH THE REGISTER.
1638
1639  006326  012701  140000          KC1:    MOV     #TESTR1,R1      ;MAKE THIS CODE MISSES
1640  006332  012702  142000                  MOV     #TESTR2,R2      ;TO BOTH GROUPS!
1641  006336  012700  001000                  MOV     #1000,R0
1642  006342  012737  000030  177746  1$:     MOV     #SOM1,@#CONTRL
1643  006350  005721                          TST     (R1)+
1644  006352  012737  000044  177746          MOV     #S1M0,@#CONTRL
1645  006360  005722                          TST     (R2)+
1646  006362  077011                          SOB     R0,1$
1647
1648  006364  017702  000352                  MOV     @KCPTR,R2       ;GET THE HIT/MISS PATTERN
1649  006370  012700  006452                  MOV     #KC3,R0         ;AND MAKE THE INSTRUCTIONS
1650  006374  012701  000007                  MOV     #7,R1           ;BETWEEN KC3 AND KC9
```

```
1651  006400  013737  006736  177746          MOV     KCCON,@#CONTRL  ;HITS AND MISSES SO THAT
1652  006406  000403                          BR      KC2.5           ;WHEN THAT CODE IS EXECUTED
1653  006410  006302                  KC2:    ASL     R2              ;THIS PATTERN WILL BE FLOATED
1654  006412  103001                          BCC     KC2.5           ;THROUGH THE HIT/MISS REGISTER.
1655  006414  005710                          TST     (R0)            ;MAKE (R0) A HIT!
1656  006416  062700  000002          KC2.5:  ADD     #2,R0
1657  006422  006302                          ASL     R2
1658  006424  103001                          BCC     1$
1659  006426  005710                          TST     (R0)            ;MAKE (R0) A HIT!
1660  006430  062700  000006          1$:     ADD     #6,R0
1661  006434  077113                          SOB     R1,KC2
1662
1663  006436  012705  177752                  MOV     #HITMIS,R5      ;NOW THAT THE HITS
1664  006442  000403                          BR      KC3             ;AND MISSES HAVE BEEN
1665                                                                  ;APPROPRIATELY ESTABLISHED
1666                                                                  ;EXECUTE THE CODE AND
1667                                                                  ;CAUSE THE PATTERN TO FLOAT
1668                                                                  ;THROUGH THE HIT/MISS
1669                                                                  ;REGISTER.
1670
1671
1672          006444                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
1673          006444                          LOC=-4&LOC
1674          006450                          LOC=LOC+4
1675          006450                          .=LOC
1676
1677  006450  000000                          HALT
1678  006452  000240                  KC3:    NOP                     ;THE HALT'S HERE ARE NOT
1679  006454  000402                          BR      KC4             ;EXECUTED, THEY ARE FILLERS.
1680  006456  000000                          HALT                    ;THE ADDRESS OF THE HIT AND
1681  006460  000000                          HALT                    ;MISS REGISTER IS IN R5.
1682  006462  011500                  KC4:    MOV     (R5),R0         ;NOTE THAT THE HIT/MISS
1683  006464  000402                          BR      KC5             ;REGISTER IS READ EVERY
1684  006466  000000                          HALT                    ;TWO CYCLES AND SAVED IN
1685  006470  000000                          HALT                    ;A PROCESSOR GENERAL
1686  006472  011501                  KC5:    MOV     (R5),R1         ;PURPOSE REGISTER.
1687  006474  000402                          BR      KC6
1688  006476  000000                          HALT
1689  006500  000000                          HALT
1690  006502  011502                  KC6:    MOV     (R5),R2
1691  006504  000402                          BR      KC7
1692  006506  000000                          HALT
1693  006510  000000                          HALT
1694  006512  011503                  KC7:    MOV     (R5),R3
1695  006514  000402                          BR      KC8
1696  006516  000000                          HALT
1697  006520  000000                          HALT
1698  006522  011504                  KC8:    MOV     (R5),R4
1699  006524  000402                          BR      KC9
1700  006526  000000                          HALT
1701  006530  000000                          HALT
1702  006532  011505                  KC9:    MOV     (R5),R5         ;CAN SAVE PATTERN IN R5
1703                                                                  ;SINCE THE ADDRESS IS
1704                                                                  ;NO LONGER NEEDED.
1705  006534  042700  177774          KC10:   BIC     #177774,R0      ;GET THE PATTERNS READ
1706  006540  010037  006770                  MOV     R0,KCR0         ;FROM THE HIT/MISS REGISTER
```

```
1707  006544  042701  017760              BIC     #17760,R1       ;INTO LOCATIONS KCR0
1708  006550  010137  006772              MOV     R1,KCR1         ;THROUGH KCR5 SO THE
1709  006554  010237  006774              MOV     R2,KCR2         ;GENERAL PURPOSE REGISTERS
1710  006560  010337  006776              MOV     R3,KCR3         ;CAN BE USED FOR OTHER
1711  006564  010437  007000              MOV     R4,KCR4         ;THINGS
1712  006570  010537  007002              MOV     R5,KCR5
1713
1714  006574  017701  000142      KC11:   MOV     @KCPTR,R1
1715  006600  005000                      CLR     R0
1716  006602  012702  000006              MOV     #6,R2           ;PUT THE EXPECTED VALUES
1717  006606  012703  007004              MOV     #KCE0,R3        ;IN KCE0 THROUGH KCE5!
1718  006612  073027  000002      KC12:   ASHC    #2,R0
1719  006616  042700  177700              BIC     #177700,R0
1720  006622  010023                      MOV     R0,(R3)+
1721  006624  077206                      SOB     R2,KC12
1722
1723  006626  012700  006770              MOV     #KCR0,R0
1724  006632  012701  007004              MOV     #KCE0,R1        ;MAKE SURE THE PATTERNS
1725  006636  012702  000006              MOV     #6,R2           ;WHICH WERE READ FROM
1726  006642  022021          KC13:       CMP     (R0)+,(R1)+     ;THE HIT AND MISS REGISTER
1727  006644  001402                      BEQ     KC14            ;MATCH THE EXPECTED
1728  006646  000137  007020              JMP     KCERR           ;PATTERNS.
1729  006652  077205          KC14:       SOB     R2,KC13
1730
1731  006654  062737  000002  006742 KC15:   ADD     #2,KCPTR                ;MOVE POINTER TO NEXT
1732  006662  023727  006742  006766      CMP     KCPTR,#KCTBLB   ;PATTERN AND IF ALL THE
1733  006670  001402                      BEQ     1$              ;PATTERNS HAVEN'T BEEN
1734  006672  000137  006326              JMP     KC1             ;TESTED GO TO KC1 TO TEST
1735                                                              ;THIS NEXT PATTERN.
1736  006676  005337  006740      1$:     DEC     KCFLG1          ;IF ALL THE PATERNS HAVE BEEN
1737  006702  100002                      BPL     KC16            ;TESTED WITH THAT GROUP CONFIGURATION
1738  006704  000137  007050              JMP     KCDONE          ;SO GO TO THE NEXT CONFIGURATION.
1739                                                              ;OR DONE!!
1740  006710  001405          KC16:       BEQ     KC17
1741  006712  012737  000044  006736      MOV     #S1MD,KCCON     ;BOTH GROUPS ENABLED CONFIGURATION
1742  006720  000137  006312              JMP     KC0             ;HAS BEEN TESTED SO NOW TEST GROUP
1743                                                              ;ZERO DISABLED CONFIGURATION.
1744  006724  012737  000030  006736 KC17:   MOV     #S0M1,KCCON     ;BOTH GROUPS ENABLED AND GROUP ZERO
1745                                                              ;DISABLED CONFIGURATIONS HAVE BOTH
1746                                                              ;BEEN TESTED SO FINALLY TEST THE
1747  006732  000137  006312              JMP     KC0             ;GROUP ONE DISABLED CONFIGURATION.
1748
1749
1750  006736  000000          KCCON:  .WORD   0               ;PATTERN BEING USED IN THE CONTROL REGISTER
1751
1752  006740  000000          KCFLG1: .WORD   0               ;FLAG USED TO DETERMINE THE CONFIGURATION
1753                                                          ;BEING TESTED.
1754  006742  000000          KCPTR:  .WORD   0               ;POINTER USED TO POINT TO THE PATTERN
1755                                                          ;BEING TESTED IN KCTBL.
1756
1757  006744  000000          KCTBL:  .WORD   0               ;PATTERNS WHICH ARE
1758  006746  002000                  .WORD   002000          ;FLOATED THROUGH THE HIT/MISS
1759  006750  177760                  .WORD   177760          ;REGISTER. ONLY THE UPPER
1760  006752  175760                  .WORD   175760          ;12 BITS HAVE ANY SIGNIFICANCE!!
1761  006754  125240                  .WORD   125240
1762  006756  146300                  .WORD   146300
```

```
1763  006760  161600                          .WORD   161600
1764  006762  100020                          .WORD   100020
1765  006764  077740                          .WORD   077740
1766  006766  000000              KCTBLB:     .WORD   0
1767
1768  006770  000000              KCR0:       .WORD   0           ;STORAGE FOR THE PATTERNS READ
1769  006772  000000              KCR1:       .WORD   0           ;OUT OF THE HUIT/MISS REGISTER.
1770  006774  000000              KCR2:       .WORD   0
1771  006776  000000              KCR3:       .WORD   0
1772  007000  000000              KCR4:       .WORD   0
1773  007002  000000              KCR5:       .WORD   0
1774
1775  007004  000000              KCE0:       .WORD   0           ;EXPECTED VALUES FOR THE PATTERNS
1776  007006  000000              KCE1:       .WORD   0           ;READ FROM THE HIT/MISS REGISTER.
1777  007010  000000              KCE2:       .WORD   0
1778  007012  000000              KCE3:       .WORD   0
1779  007014  000000              KCE4:       .WORD   0
1780  007016  000000              KCE5:       .WORD   0
1781
1782  007020                      KCERR:                          ;REPORT THE PATTERN READ FROM THE
1783  007020  013737  006736  001236          MOV     KCCON,$TMP2 ;HIT/MISS REGISTER WAS NOT THE EXPECTED
1784  007026  104120              1$:         ERROR   120         ;VALUE.
1785  007030  012737  177777  031100          MOV     #-1,CONFL2
1786  007036  012737  177777  031104          MOV     #-1,HIMFL2
1787  007044  000137  006654                  JMP     KC15
1788
1789  007050  005037  177746      KCDONE:     CLR     @#CONTRL    ;DONE!!
1790
1791                              ;;**********************************************************************
1792                              ;*TEST 12       CACHE CONTROL AND HIT/MISS REGISTERS EVALUATION ROUTINE
1793                              ;*
1794                              ;*THIS IS NOT A TEST. THIS ROUTINE IS USED TO LOOK AT THE RESULTS
1795                              ;*OF TST5 THROUGH TST10, WHICH TESTED THE HIT/MISS REGISTER
1796                              ;*AND THE CONTROL REGISTER. THOSE TESTS HAVE SIGNALLED A BAD
1797                              ;*REGISTER USING THE FLAGS, CONFL2 AND HIMFL2, REPRESENTING THE
1798                              ;*CONTROL AND HIT/MISS REGISTERS RESPECTIVELY. IF ONE OF THESE
1799                              ;*REGISTERS WAS FOUND TO BE BAD THE FLAG SHOULD BE A -1. WHILE A
1800                              ;*ZERO FLAG INDICATES THAT THOSE TESTS FOUND THAT REGISTER
1801                              ;*FUNCTIONAL. THIS ROUTINE LOOKS AT THE FLAGS, CONFL2 AND HIMFL2,
1802                              ;*WHICH ARE CONSIDERED TO BE LOCAL AND TRANSFERS THE INDICATORS
1803                              ;*THEY CONTAIN TO THE GLOBAL FLAGS, CONFLG AND HIMFLG. THESE GLOBAL
1804                              ;*FLAGS ARE USED TO DESIGNATE TO THE REST OF THE PROGRAM THE FUNCTIONALITY
1805                              ;*OR DISFUNCTIONALITY OF THOSE REGISTERS.
1806                              ;*
1807                              ;;**********************************************************************
1808  007054  000004              TST12:  SCOPE
1809          000012              KY=$TN-1
1810  007056  005737  031100              TST     CONFL2
1811  007062  001403                      BEQ     KY1
1812  007064  012737  177777  031064      MOV     #-1,CONFLG
1813  007072  005737  031104      KY1:    TST     HIMFL2
1814  007076  001403                      BEQ     KY2
1815  007100  012737  177777  031070      MOV     #-1,HIMFLG
1816  007106                      KY2:                            ;DONE
1817
1818                              ;;**********************************************************************
```

```
                                        ;*TEST 13       CACHE CONTROL LOGIC, 'RANDOM' FLIP FLOP TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE 'RANDOM' CONTROL SIGNAL.
                                        ;*A TEST IS MADE TO INSURE THAT THE 'RANDOM' FLIP-FLOP IS NOT STUCK
                                        ;*AND IS TOGGLED ONCE FOR EVERY 'BUST' CYCLE INITIATED BY
                                        ;*THE PROCESSOR. 'BUST' IS BUS START, A SIGNAL PRODUCED BY
                                        ;*THE PROCESSOR WHENEVER IT THINKS IT IS ABOUT TO DO A MEMORY CYCLE.
                                        ;*THE RANDOM FLIP FLOP IS USED IN THE CACHE TO DETERMINE WHICH
                                        ;*GROUP TO WRITE IN THE EVENT OF A READ MISS CYCLE. IF THIS FLIP FLOP IS
                                        ;*SET THEN GROUP ZERO IS WRITTEN; IF CLEAR THEN GROUP ONE IS WRITTEN.
                                        ;*
                                        ;;*****************************************************************
007106  000004                        TST13:  SCOPE
007110  012737  000040  001302                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000013                  KF=$TN-1
                                                                        ;SET THE SKAD REGISTER
007116  012737  007342  030646          MOV     #TST14,SKAD     ;IN CASE THE TEST ABORTS.

007124  113737  001102  001232          MOVB    $TSTNM,$TMP0
007132  012737  030522  000114          MOV     #SPUR,@#CACHVEC ;EXPECT NO PARITY ERRORS.

007140  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
007142  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
007144  012700  007340          KF1:    MOV     #KFTMP2,R0      ;ESTABLISH A LOCATION FOR THE
                                                                ;HITS TO BE MADE WHICH WON'T
                                                                ;INTERFER WITH THE HITS CAUSED
                                                                ;BY EXECUTION OF THIS CODE!
007150  042700  176003                  BIC     #176003,R0
007154  010001                          MOV     R0,R1
007156  062701  140000                  ADD     #TESTR1,R1
007162  010002                          MOV     R0,R2
007164  062702  142000                  ADD     #TESTR2,R2

007170  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;MAKE THOSE TWO TEST LOCATIONS
007176  005710                          TST     (R0)            ;(R1) AND (R2) MISSES IN BOTH
                                                                ;GROUPS BY MAKING (R0) A HIT
                                                                ;IN BOTH GROUPS.

007200  005710                          TST     (R0)

                                                                ;SEE IF REFERENCE ADDRESS
007202  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
007210  001006                          BNE     KF2

                                                                ;IF NOT ERROR!
007212  010037  001236                  MOV     R0,$TMP2
007216  012737  000001  001234          MOV     #1,$TMP1
007224  104001                          ERROR   1


007226  012737  000030  177746  KF2:    MOV     #S0M1,@#CONTRL
007234  005710                          TST     (R0)

007236  005710                          TST     (R0)
```

```
                                                                ;SEE IF REFERENCE ADDRESS
007240  032737  000010  177752          BIT     #10,@#HITMIS  ;IS A HIT.
007246  001006                        BNE     KF3
                                                                ;IF NOT ERROR!
007250  010037  001236                MOV     R0,$TMP2
007254  012737  000000  001234        MOV     #0,$TMP1
007262  104001                        ERROR   1
007264  005037  177746          KF3:  CLR     @#CONTRL        ;NOW THAT THE ADDRESSES (R1)
                                                                ;AND (R2) ARE MISSES, REFERENCING
                                                                ;THEM BOTH EACH IN CONSECUTIVE
                                                                ;REFERNCES SHOULD CAUSE THEM BOTH
                                                                ;TO BE MADE HITS IF THE RANDOM
                                                                ;FLIP FLOP TOGGLES INBETWEEN THE
                                                                ;TWO CYCLES!
                                                                ;NOTE THAT THESE TWO ADDRESSES
                                                                ;(R1) AND (R2) ARE SUCH THAT
                                                                ;IF THE RANDOM FLIP FLOP DIDN'T TOGGLE
                                                                ;THE HITS AT THE ADDRESSES
                                                                ;WOULD BE MUTUALLY EXCLUSIVE,
                                                                ;THAT IS BOTH THESE ADDRESSES
                                                                ;CAN'T BE HITS IN THE SAME GROUP!
007270  000240                        NOP                     ;FOR SCOPING WITH AN OSCILLOSCOPE!
007272  021112                        CMP     (R1),(R2)       ;HERE BOTH THE OPERAND FETCHES
                                                                ;SHOULD BE MISSES.
007274  021112                        CMP     (R1),(R2)       ;HERE BOTH THE OPERAND FETCHES
                                                                ;SHOULD BE HITS!
007276  013705  177752                MOV     @#HITMIS,R5
007302  005105                        COM     R5
007304  032705  000014                BIT     #14,R5          ;BOTH HITS ELSE ERROR.
007310  001411                        BEQ     KF4
007312  010137  001236                MOV     R1,$TMP2        ;REPORT THE ERROR.
007316  005037  001240                CLR     $TMP3
007322  010237  001242                MOV     R2,$TMP4
007326  005037  001244                CLR     $TMP5
007332  104121                1$:     ERROR   121
007334  000402                KF4:    BR      KF5
007336  000000                KFTMP1: .WORD   0               ;USED TO DETERMINE THE TEST
007340  000000                KFTMP2: .WORD   0               ;ADDRESSES.
007342                        KF5:                            ;DONE!
                                ;;*****************************************************
                                ;*TEST 14       CACHE MAINTENANCE REGISTER COUNT PATTERN TEST
                                ;*
                                ;*THIS TEST RUNS A COUNT PATTERN THROUGH THE MAINTENANCE REGISTER'S
                                ;*BITS 15 TO 4. THIS IS DONE TO INSURE THAT THESE BITS ARE SETABLE
                                ;*AND THAT THE DATA PATH TO THE REGISTERS IS VIABLE. MISSES ARE FORCED
                                ;*TO BOTH GROUPS SO THAT NO CACHE DATA OR ADDRESS MEMORY
```

```
                                        ;*ERRORS SHOULD OCCUR. ALSO ANY CYCLES DONE TO MAIN MEMORY
                                        ;*ARE INSURED, BY PROPER SELECTION OF INSTRUCTIONS, TO RETURN
                                        ;*DATA WITH THE PARITY BITS ON SO AS TO NOT CAUSE MAIN MEMORY PARITY
                                        ;*ERRORS BY SETTING THE MAIN MEMORY MAINTENANCE FUNCTION WHICH WOULD
                                        ;*EFFECTIVELY FORCE THE PARITY BITS READ FROM MAIN MEMORY TO A
                                        ;*ONE. SINCE THESE PARITY ARE ALREADY ONES, NO ERRORS SHOULD OCCUR.
                                        ;*
                                        ;;*****************************************************************
007342  000004                  $T14:   SCOPE
007344  012737  000020  001302          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000014                  MA=$TN-1
                                                                ;SET THE SKAD REGISTER
007352  012737  007624  030646          MOV     #TST15,SKAD     ;IN CASE THE TEST ABORTS.

007360  113737  001102  001232          MOVB    $TSTNM,$TMPO

007366  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
007370  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
007372  012737  007526  000114          MOV     #MAERR,@#CACHVEC        ;IN CASE AN ERROR OCCURS WHILE
                                                                ;RUNNING A COUNT PATTERN
                                                                ;THROUGH THE MAINTENANCE
                                                                ;REGISTER SET UP THE PARITY ERROR
                                                                ;TRAP VECTOR; NOTE THAT NO ERRORS
                                                                ;SHOULD OCCUR IF THIS REGISTER
                                                                ;AND THE PARITY LOGIC IS FUNCTIONING
                                                                ;PROPERLY!
007400  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.

007406  012701  177750                  MOV     #MAINT,R1
007412  005004                          CLR     R4
007414  012737  007426  001110          MOV     #MA1,$LPERR
007422  012700  170000                  MOV     #170000,R0

007426  000240                  MA1:    NOP                     ;NOTE, THE CODE IN THIS ARE
007430  010411                          MOV     R4,(R1)         ;MA1 THROUGH MA2, ASSEMBLES TO
007432  011102                          MOV     (R1),R2         ;MACHINE CODE WHICH WILL
007434  005011                          CLR     (R1)            ;HAVE THE PARITY BITS ON, 1'S!
                                                                ;THE PATTERN IS LOADDED INTO THE
                                                                ;MAINETENANCE REGISTER, READ BACK
                                                                ;AND THE MAINTENANCE REGISTER
                                                                ;IS CLEARED.
007436  030011                          BIT     R0,(R1)         ;SEE IF ANY OF THE HIGH ORDER
                                                                ;FOUR BITS, 15 TO 12,
                                                                ;THE BITS WHICH CONTROL THE
                                                                ;MAIN MEMORY DATA PARITY MAINTENANCE
                                                                ;FUNCTION ARE STUCK ON.
007440  001402                          BEQ     .+6             ;IF SO, THEN ALL THAT CAN
007442  000000                          HALT                    ;BE DONE IS TO HALT!!!!!
                                                                ;FOR IF CONTROL IS PASSED TO
                                                                ;ANY OTHER PART OF THIS PROGRAM
                                                                ;THERE WOULD BE NO CONTROL
                                                                ;OVER WHAT KIND OF DATA WOULD
                                                                ;BE READ FROM MAIN MEMORY AND
                                                                ;MAIN MEMORY DATA PARITY ERRORS
                                                                ;WOULD BE LIKELY TO OCCUR.
007444  000240                  MA2:    NOP
```

```
1987
1988  007446  011105                          MOV     (R1),R5         ;SEE IF ANY OF THE LOW ORDER
1989  007450  001410                          BEQ     MA3             ;BITS, 11 THROUGH 0, ARE STUCK
1990                                                                  ;AT ONE.
1991                                                                  ;IF SO REPORT THE ERROR.
1992  007452  010437  001236                  MOV     R4,$TMP2
1993  007456  010537  001240                  MOV     R5,$TMP3
1994  007462  104122                  1$:     ERROR   122
1995  007464  012737  177777  031066          MOV     #-1,MANFLG      ;??????????????GO ON???????????
1996
1997  007472  020402                  MA3:    CMP     R4,R2           ;SEE IF THE PATTERN WRITTEN MATCHES
1998  007474  001410                          BEQ     MA4             ;THE PATTERN READ.
1999
2000                                                                  ;IF NOT REPORT THE ERROR.
2001  007476  010437  001236                  MOV     R4,$TMP2
2002  007502  010237  001240                  MOV     R2,$TMP3
2003  007506  104123                  1$:     ERROR   123
2004  007510  012737  177777  031102          MOV     #-1,MANFL2
2005
2006  007516  062704  000020          MA4:    ADD     #20,R4          ;INCREMENT THE COUNT PATTERN.
2007  007522  001341                          BNE     MA1
2008  007524  000432                          BR      MADONE
2009
2010  007526                          MAERR:                          ;TRAP TO HERE IN THE EVENT
2011                                                                  ;THAT A PARITY ERROR OCCURS
2012                                                                  ;WHILE RUNNING THIS COUNT
2013                                                                  ;PATTERN TEST.
2014  007526  032737  000400  177744          BIT     #400,@#MEMERR   ;SEE IF THE ERROR WAS A MAINTENANCE
2015  007534  001005                          BNE     MAERR1          ;ERROR, CAUSED BY A MAINTENANCE
2016                                                                  ;FUNCTION. IF NOT GO TO THE
2017  007536  012737  030522  000114          MOV     #SPUR,@#CACHVEC ;SPUR ROUTINE WHICH HANDLES SUCH UNEXPECTED
2018  007544  000137  030522                  JMP     SPUR            ;ERRORS.
2019
2020  007550  013737  177744  001242  MAERR1: MOV     @#MEMERR,$TMP4  ;IF THE ERROR WAS CAUSED BY A
2021  007556  013737  177740  001234          MOV     @#LOADRS,$TMP1  ;MAINT FUNCTION THEN REPORT THE
2022  007564  013737  177742  001236          MOV     @#HIADRS,$TMP2  ;FAILURE OF THAT REGISTER.
2023  007572  012637  001240                  MOV     (SP)+,$TMP3
2024  007576  005726                          TST     (SP)+
2025  007600  104124                  1$:     ERROR   124
2026  007602  012737  177777  031102          MOV     #-1,MANFL2
2027
2028  007610  000742                          BR      MA4             ;RETURN TO THE TEST.
2029
2030  007612  005037  177746  MADONE: CLR     @#CONTRL                ;DONE
2031  007616  012737  030522  000114          MOV     #SPUR,@#CACHVEC
2032
2033
2034
2035
2036                                  ;;*****************************************************************
2037                                  ;*TEST 15       CACHE MAINTENANCE AND ERROR REGISTERS TEST 1
2038                                  ;*
2039                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE A PARITY
2040                                  ;*ERROR ON THE MAIN MEMORY ADDRESS AND CONTROL LINES, AND ALSO A TEST
2041                                  ;*OF THE ERROR REGISTER'S ABILITY TO APPROPRIATELY SET TO 104402. THE
2042                                  ;*REFERENCE CAUSING THIS ERROR WILL BE MADE FROM THE CPU DIRECTLY TO
```

```
                                        ;*THE CACHE.
                                        ;*
                                        ;;*****************************************************************
007624  000004                        TST15:  SCOPE
007626  012737  000040  001302                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000015                  MAB=$TN-1
                                                                        ;SET THE SKAD REGISTER
007634  012737  010122  030646                MOV     #TST16,SKAD     ;IN CASE THE TEST ABORTS.

007642  113737  001102  001232                MOVB    $TSTNM,$TMPO

007650  104415                                SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
007652  104416                                SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
007654  104417                                SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
007656  104420                                SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
007660  012737  007730  000114                MOV     #MABRR0,@#CACHVEC       ;SET UP FOR THE ERROR.

007666  012704  000002                        MOV     #2,R4           ;THIS IS THE PATTERN THAT WILL
007672  012702  177750                        MOV     #MAINT,R2       ;BE PUT IN THE MAINTENANCE REG.
007676  012737  000014  177746                MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.

007704  000240                                NOP                     ;FOR SCOPING.
007706  010412                                MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
007710  005012                                CLR     (R2)            ;THE REFERENCE WHICH FETCHES
                                                                        ;THIS INSTRUCTION SHOULD
                                                                        ;CAUSE THE ABORT!

007712                          MAB2:                                 ;NO ABORT OCCURRED REPORT THE ERROR
007712  010437  001236                1$:     MOV     R4,$TMP2
007716  104127                                ERROR   127
007720  012737  177777  031102                MOV     #-1,MANFL2
007726  000474                                BR      MABDON

007730  022737  104402  177744  MABRR0:       CMP     #104402,@#MEMERR        ;WHEN THE TRAP IS MADE TO THIS LOCATION
007736  001036                                BNE     MABRR4          ;MAKE SURE THE ERROR REGISTER IS
                                                                        ;SET CORRECTLY. IF NOT GO TO MABRR4.
007740  022626                  MABRR1:       CMP     (SP)+,(SP)+     ;OTHERWISE RESET THE STACK.
007742  012737  177777  177744  MABR15:       MOV     #-1,@#MEMERR    ;ATTEMPT TO CLEAR THE ERROR REGISTER.
007750  005737  177744                        TST     @#MEMERR
007754  001416                                BEQ     MABRR3

007756                          MABRR2:                               ;REPORT ERROR REGISTER WON'T CLEAR!
007756  013737  177740  001236                MOV     @#LOADRS,$TMP2
007764  013737  177742  001240                MOV     @#HIADRS,$TMP3
007772  013737  177744  001242                MOV     @#MEMERR,$TMP4
010000  104130                1$:     ERROR   130
010002  012737  177777  031062                MOV     #-1,MMRFLG
010010  000443                                BR      MABDON

010012  022737  177740  177740  MABRR3:       CMP     #177740,@#LOADRS        ;MAKE SURE THE ADDRESS
010020  001356                                BNE     MABRR2          ;REGISTER RESET.
010022  022737  000003  177742                CMP     #3,@#HIADRS
010030  001352                                BNE     MABRR2
010032  000432                                BR      MABDON

010034                          MABRR4:                               ;REPORT ERROR REGISTER NOT SET CORRECTLY!!
```

```
2099  010034  012637  001236                  MOV     (SP)+,$TMP2
2100  010040  005726                          TST     (SP)+
2101  010042  013737  177740  001240          MOV     @#LOADRS,$TMP3
2102  010050  013737  177742  001242          MOV     @#HIADRS,$TMP4
2103  010056  012737  000002  001244          MOV     #2,$TMP5
2104  010064  012737  104402  001246          MOV     #104402,$TMP6
2105  010072  013737  177744  001250          MOV     @#MEMERR,$TMP7
2106  010100  104131                  1$:     ERROR   131
2107  010102  012737  177777  031102          MOV     #-1,MANFL2
2108  010110  012737  177777  031076          MOV     #-1,MMRFL2
2109  010116  000711                          BR      MABR15          ;GO SEE IF THE ERROR REGISTER
2110                                                                  ;CAN BE CLEARED.
2111  010120  104410                  MABDON: RSET                    ;DONE!!
2112
2113
2114                                  ;;**************************************************************
2115                                  ;*TEST 16       CACHE MAINTENANCE AND ERROR REGISTERS TEST 2
2116                                  ;*
2117                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2118                                  ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE,
2119                                  ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2120                                  ;*
2121                                  ;;**************************************************************
2122  010122  000004                  TST16:  SCOPE
2123  010124  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2124           000016                 MB=$TN-1
2125                                                                  ;SET THE SKAD REGISTER
2126  010132  012737  010440  030646          MOV     #TST17,SKAD     ;IN CASE THE TEST ABORTS.
2127
2128  010140  113737  001102  001232          MOVB    $TSTNM,$TMPO
2129
2130  010146  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2131  010150  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2132  010152  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2133  010154  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2134  010156  012737  010236  000114          MOV     #MBERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
2135  010164  012704  010000                  MOV     #10000,R4       ;PATERN TO BE PUT INTO THE
2136  010170  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
2137  010174  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
2138  010202  000402                          BR      MB1
2139
2140           010204                         LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2141           010204                         LOC=-4&LOC
2142           010210                         LOC=LOC+4
2143           010210                         .=LOC
2144
2145  010210  000240                  MB1:    NOP
2146  010212  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
2147  010214  005701                  MB2:    TST     R1              ;THIS IS A DUMMY INSTRUCTION
2148                                                                  ;WITH THE APPROPRIATE PARITY
2149                                                                  ;WHOSE FETCH WILL CAUSE THE ERROR.
2150  010216  005012                          CLR     (R2)
2151
2152  010220                          MB3:                            ;REPORT ERROR.  MAINTENANCE
2153  010220  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2154                                                                  ;CAUSE ERROR.
```

```
2155  010224  104127                    1$:     ERROR   127
2156  010226  012737  177777  031102    MOV     #-1,MANFL2
2157  010234  000500                    BR      MBDONE
2158
2159  010236  022737  104404  177744    MBERRO: CMP     #104404,@#MEMERR        ;DID THE ERROR REGISTER
2160  010244  001042                    BNE     69$             ;SET PROPERLY?
2161
2162  010246  022626                    64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2163  010250  005037  177572            65$:    CLR     @#MMR0
2164  010254  005037  172516            CLR     @#MMR3
2165  010260  012737  177777  177744    MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2166  010266  005737  177744            TST     @#MEMERR        ;REGISTER.
2167  010272  001416                    BEQ     68$
2168
2169  010274                            66$:                    ;ERROR REGISTER WON'T
2170  010274  013737  177740  001236    MOV     @#LOADRS,$TMP2  ;CLEAR
2171  010302  013737  177742  001240    MOV     @#HIADRS,$TMP3
2172  010310  013737  177744  001242    MOV     @#MEMERR,$TMP4
2173
2174  010316  104130                    67$:    ERROR   130
2175  010320  012737  177777  031062    MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2176  010326  000443                    BR      MBDONE
2177
2178  010330  022737  177740  177740    68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2179  010336  001356                    BNE     66$             ;UNLOCKED.
2180  010340  022737  000003  177742    CMP     #3,@#HIADRS
2181  010346  001352                    BNE     66$
2182  010350  000432                    BR      MBDONE
2183
2184  010352                            69$:                    ;REPORT ERROR REGISTER
2185  010352  012637  001236            MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2186  010356  005726                    TST     (SP)+           ;RESET THE STACK.
2187  010360  013737  177740  001240    MOV     @#LOADRS,$TMP3
2188  010366  013737  177742  001242    MOV     @#HIADRS,$TMP4
2189  010374  012737  010000  001244    MOV     #10000,$TMP5
2190  010402  012737  104404  001246    MOV     #104404,$TMP6
2191  010410  013737  177744  001250    MOV     @#MEMERR,$TMP7
2192
2193  010416  104131                    70$:    ERROR   131
2194  010420  012737  177777  031102    MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2195  010426  012737  177777  031076    MOV     #-1,MMRFL2
2196  010434  000705                    BR      65$
2197  010436  104410                    MBDONE: RSET
2198
2199                                    ;;*****************************************************************
2200                                    ;*TEST 17        CACHE MAINTENANCE AND ERROR REGISTERS TEST 3
2201                                    ;*
2202                                    ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2203                                    ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S HIGH BYTE,
2204                                    ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2205                                    ;*
2206                                    ;;*****************************************************************
2207  010440  000004                    TST17:  SCOPE
2208  010442  012737  000040  001302            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2209          000017                    MC=$TN-1
2210                                                                    ;SET THE SKAD REGISTER
```

```
2211  010450  012737  010754  030646          MOV     #TST20,SKAD     ;IN CASE THE TEST ABORTS.
2212
2213  010456  113737  001102  001232          MOVB    $TSTNM,$TMP0
2214
2215  010464  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2216  010466  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2217  010470  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2218  010472  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2219  010474  012737  010552  000114          MOV     #MCERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
2220  010502  012704  020000                  MOV     #20000,R4       ;PATTERN TO BE USED IN THE
2221  010506  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
2222  010512  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
2223  010520  000401                          BR      MC1
2224
2225          010522                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2226          010520                          LOC=-4&LOC
2227          010524                          LOC=LOC+4
2228          010524                          .=LOC
2229
2230  010524  000240                  MC1:    NOP
2231  010526  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
2232  010530  005701                  MC2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
2233                                                                  ;SHOULD CAUSE THE ABORT.
2234  010532  005012                          CLR     (R2)
2235
2236  010534                          MC3:                            ;REPORT ERROR.  MAINTENANCE
2237  010534  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2238                                                                  ;CAUSE ERROR.
2239  010540  104127                  1$:     ERROR   127
2240  010542  012737  177777  031102  MOV     #-1,MANFL2
2241  010550  000500                          BR      MCDONE
2242
2243  010552  022737  104404  177744  MCERR0: CMP     #104404,@#MEMERR        ;DID THE ERROR REGISTER
2244  010560  001042                          BNE     69$             ;SET PROPERLY?
2245
2246  010562  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2247  010564  005037  177572          65$:    CLR     @#MMR0
2248  010570  005037  172516                  CLR     @#MMR3
2249  010574  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2250  010602  005737  177744                  TST     @#MEMERR        ;REGISTER.
2251  010606  001416                          BEQ     68$
2252
2253  010610                          66$:                            ;ERROR REGISTER WON'T
2254  010610  013737  177740  001236          MOV     @#LOADRS,$TMP2  ;CLEAR
2255  010616  013737  177742  001240          MOV     @#HIADRS,$TMP3
2256  010624  013737  177744  001242          MOV     @#MEMERR,$TMP4
2257
2258  010632  104130                  67$:    ERROR   130
2259  010634  012737  177777  031062          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2260  010642  000443                          BR      MCDONE
2261
2262  010644  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2263  010652  001356                          BNE     66$             ;UNLOCKED.
2264  010654  022737  000003  177742          CMP     #3,@#HIADRS
2265  010662  001352                          BNE     66$
2266  010664  000432                          BR      MCDONE
```

```
2267
2268  010666                                  69$:                                    ;REPORT ERROR REGISTER
2269  010666  012637  001236                          MOV     (SP)+,$TMP2             ;NOT SET AS EXPECTED.
2270  010672  005726                                  TST     (SP)+                   ;RESET THE STACK.
2271  010674  013737  177740  001240                  MOV     @#LOADRS,$TMP3
2272  010702  013737  177742  001242                  MOV     @#HIADRS,$TMP4
2273  010710  012737  020000  001244                  MOV     #20000,$TMP5
2274  010716  012737  104404  001246                  MOV     #104404,$TMP6
2275  010724  013737  177744  001250                  MOV     @#MEMERR,$TMP7
2276
2277  010732  104131                          70$:    ERROR   131
2278  010734  012737  177777  031102                  MOV     #-1,MANFL2              ;SIGNAL BAD REGISTER
2279  010742  012737  177777  031076                  MOV     #-1,MMRFL2
2280  010750  000705                                  BR      65$
2281  010752  104410                          MCDONE: RSET
2282
2283                                          ;;**************************************************************
2284                                          ;*TEST 20        CACHE MAINTENANCE AND ERROR REGISTERS TEST 4
2285                                          ;*
2286                                          ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2287                                          ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE,
2288                                          ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2289                                          ;*
2290                                          ;;**************************************************************
2291  010754  000004                          TST20:  SCOPE
2292  010756  012737  000040  001302                  MOV     #40,$TIMES              ;;DO 40 ITERATIONS
2293          000020                          MD=$TN-1
2294                                                                                  ;SET THE SKAD REGISTER
2295  010764  012737  011274  030646                  MOV     #TST21,SKAD             ;IN CASE THE TEST ABORTS.
2296
2297  010772  113737  001102  001232                  MOVB    $TSTNM,$TMP0
2298
2299  011000  104415                                  SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2300  011002  104416                                  SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2301  011004  104417                                  SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2302  011006  104420                                  SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2303  011010  012737  011072  000114                  MOV     #MDERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
2304  011016  012704  040000                          MOV     #40000,R4               ;PATTERN TO BE PUT IN THE
2305  011022  012702  177750                          MOV     #MAINT,R2               ;MAINTENANCE REGISTER.
2306  011026  012737  000014  177746                  MOV     #MDM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
2307  011034  000402                                  BR      MD1
2308
2309          011036                                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2310          011034                                  LOC=-4&LOC
2311          011040                                  LOC=LOC+4
2312          011040                                  .=LOC
2313
2314  011040  000240                                  NOP
2315  011042  000240                          MD1:    NOP
2316  011044  010412                                  MOV     R4,(R2)                 ;SET THE MAINTENANCE REGISTER.
2317  011046  005701                          MD2:    TST     R1                      ;THE FETCH OF THIS INSTRUCTION
2318                                                                                  ;SHOULD CAUSE THE MAIN MEMORY
2319                                                                                  ;DATA PARITY ABORT.
2320  011050  005012                                  CLR     (R2)
2321  011052  000240                                  NOP
2322
```

```
2323  011054                          MD3:                            ;REPORT ERROR.  MAINTENANCE
2324  011054  010437  001236                MOV     R4,$TMP2          ;FUNCTION FAILED TO
2325                                                                    ;CAUSE ERROR.
2326  011060  104127                  1$:    ERROR   127
2327  011062  012737  177777  031102  MOV     #-1,MANFL2
2328  011070  000500                  BR      MDDONE
2329
2330  011072  022737  104410  177744  MDERRO: CMP     #104410,@#MEMERR        ;DID THE ERROR REGISTER
2331  011100  001042                  BNE     69$               ;SET PROPERLY?
2332
2333  011102  022626                  64$:    CMP     (SP)+,(SP)+       ;RESET THE STACK
2334  011104  005037  177572          65$:    CLR     @#MMR0
2335  011110  005037  172516                  CLR     @#MMR3
2336  011114  012737  177777  177744          MOV     #-1,@#MEMERR      ;TRY TO CLEAR THE ERROR
2337  011122  005737  177744                  TST     @#MEMERR          ;REGISTER.
2338  011126  001416                  BEQ     68$
2339
2340  011130                          66$:                            ;ERROR REGISTER WON'T
2341  011130  013737  177740  001236          MOV     @#LOADRS,$TMP2    ;CLEAR
2342  011136  013737  177742  001240          MOV     @#HIADRS,$TMP3
2343  011144  013737  177744  001242          MOV     @#MEMERR,$TMP4
2344
2345  011152  104130                  67$:    ERROR   130
2346  011154  012737  177777  031062          MOV     #-1,MMRFLG        ;SIGNAL BAD REGISTER
2347  011162  000443                  BR      MDDONE
2348
2349  011164  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2350  011172  001356                  BNE     66$               ;UNLOCKED.
2351  011174  022737  000003  177742          CMP     #3,@#HIADRS
2352  011202  001352                  BNE     66$
2353  011204  000432                  BR      MDDONE
2354
2355  011206                          69$:                            ;REPORT ERROR REGISTER
2356  011206  012637  001236                  MOV     (SP)+,$TMP2       ;NOT SET AS EXPECTED.
2357  011212  005726                  TST     (SP)+             ;RESET THE STACK.
2358  011214  013737  177740  001240          MOV     @#LOADRS,$TMP3
2359  011222  013737  177742  001242          MOV     @#HIADRS,$TMP4
2360  011230  012737  040000  001244          MOV     #40000,$TMP5
2361  011236  012737  104410  001246          MOV     #104410,$TMP6
2362  011244  013737  177744  001250          MOV     @#MEMERR,$TMP7
2363
2364  011252  104131                  70$:    ERROR   131
2365  011254  012737  177777  031102          MOV     #-1,MANFL2        ;SIGNAL BAD REGISTER
2366  011262  012737  177777  031076          MOV     #-1,MMRFL2
2367  011270  000705                  BR      65$
2368  011272  104410                  MDDONE: RSET
2369
2370                                  ;;*****************************************************************
2371                                  ;*TEST 21       CACHE MAINTENANCE AND ERROR REGISTERS TEST 5
2372                                  ;*
2373                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2374                                  ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S HIGH BYTE,
2375                                  ;*WHEN THAT WORD IS THE WANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2376                                  ;*
2377                                  ;;*****************************************************************
2378  011274  000004                  TST21:  SCOPE
```

```
2379  011276  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2380          000021                  ME=$TN-1
2381                                                                  ;SET THE SKAD REGISTER
2382  011304  012737  011614  030646          MOV     #TST22,SKAD     ;IN CASE THE TEST ABORTS.
2383
2384  011312  113737  001102  001232          MOVB    $TSTNM,$TMPO
2385
2386  011320  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2387  011322  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2388  011324  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2389  011326  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2390  011330  012737  011412  000114          MOV     #MEERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
2391  011336  012704  100000                  MOV     #100000,R4      ;PATTERN TO BE PUT IN THE
2392  011342  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REGISTER.
2393  011346  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
2394  011354  000402                          BR      ME1
2395
2396          011356                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2397          011354                          LOC=-4&LOC
2398          011360                          LOC=LOC+4
2399          011360                          .=LOC
2400
2401  011360  000240                          NOP
2402  011362  000240                  ME1:    NOP
2403  011364  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
2404  011366  005701                  ME2:    TST     R1              ;THE FETCH OF THIS INSTRUSTION
2405                                                                  ;SHOULD CAUSE THE ABORT.
2406  011370  005012                          CLR     (R2)
2407  011372  000240                          NOP                             . . .
2408
2409  011374                          ME3:                            ;REPORT ERROR.  MAINTENANCE
2410  011374  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2411                                                                  ;CAUSE ERROR.
2412  011400  104127                  1$:     ERROR   127
2413  011402  012737  177777  031102  MOV     #-1,MANFL2
2414  011410  000500                          BR      MEDONE
2415
2416  011412  022737  104410  177744  MEERRO: CMP     #104410,@#MEMERR        ;DID THE ERROR REGISTER
2417  011420  001042                          BNE     69$             ;SET PROPERLY?
2418
2419  011422  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2420  011424  005037  177572          65$:    CLR     @#MMR0
2421  011430  005037  172516                  CLR     @#MMR3
2422  011434  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2423  011442  005737  177744                  TST     @#MEMERR        ;REGISTER.
2424  011446  001416                          BEQ     68$
2425
2426  011450                          66$:                            ;ERROR REGISTER WON'T
2427  011450  013737  177740  001236          MOV     @#LOADRS,$TMP2  ;CLEAR
2428  011456  013737  177742  001240          MOV     @#HIADRS,$TMP3
2429  011464  013737  177744  001242          MOV     @#MEMERR,$TMP4
2430
2431  011472  104130                  67$:    ERROR   130
2432  011474  012737  177777  031062          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2433  011502  000443                          BR      MEDONE
2434
```

```
2435  011504  022737  177740  177740  69$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2436  011512  001356                          BNE     66$              ;UNLOCKED.
2437  011514  022737  000003  177742          CMP     #3,@#HIADRS
2438  011522  001352                          BNE     66$
2439  011524  000432                          BR      MEDONE
2440
2441  011526                                  69$:                     ;REPORT ERROR REGISTER
2442  011526  012637  001236                  MOV     (SP)+,$TMP2      ;NOT SET AS EXPECTED.
2443  011532  005726                          TST     (SP)+            ;RESET THE STACK.
2444  011534  013737  177740  001240          MOV     @#LOADRS,$TMP3
2445  011542  013737  177742  001242          MOV     @#HIADRS,$TMP4
2446  011550  012737  100000  001244          MOV     #100000,$TMP5
2447  011556  012737  104410  001246          MOV     #104410,$TMP6
2448  011564  013737  177744  001250          MOV     @#MEMERR,$TMP7
2449
2450  011572  104131                          70$:    ERROR   131
2451  011574  012737  177777  031102          MOV     #-1,MANFL2       ;SIGNAL BAD REGISTER
2452  011602  012737  177777  031076          MOV     #-1,MMRFL2
2453  011610  000705                          BR      65$
2454  011612  104410                          MEDONE: RSET
2455
2456                                          ;;*****************************************************************
2457                                          ;*TEST 22       CACHE MAINTENANCE AND ERROR REGISTERS TEST 6
2458                                          ;*
2459                                          ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
2460                                          ;*A PARITY ERROR ON THE MAIN MEMORY EVEN WORD'S LOW BYTE,
2461                                          ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
2462                                          ;*
2463                                          ;;*****************************************************************
2464  011614  000004                          TST22:  SCOPE
2465  011616  012737  000040  001302                  MOV     #40,$TIMES       ;;DO 40 ITERATIONS
2466           000022                         MF=$TN-1
2467                                                                   ;SET THE SKAD REGISTER
2468  011624  012737  012130  030646                  MOV     #TST23,SKAD      ;IN CASE THE TEST ABORTS.
2469
2470  011632  113737  001102  001232                  MOVB    $TSTNM,$TMP0
2471  011640  012737  011726  000114                  MOV     #MFERR0,@#CACHVEC        ;SET UP FOR THE ERROR.
2472  011646  012704  010000                          MOV     #10000,R4        ;PATTERN TO BE LOADED INTO THE
2473  011652  012702  177750                          MOV     #MAINT,R2        ;MAINTENANCE REGISTER.
2474  011656  012737  000014  177746                  MOV     #MOM1,@#CONTRL   ;FORCE MISSES TO BOTH GROUPS.
2475  011664  012705  011706                          MOV     #MF2,R5          ;A REFERENCE TO THIS ADDRESS
2476                                                                           ;WILL CAUSE A PARITY TRAP BECAUSE
2477                                                                           ;THE OTHER WORD IN THE PAIR
2478                                                                           ;WILL HAVE THE APPROPRIATE
2479                                                                           ;PARITY TO CAUSE THE MAINTENANCE
2480                                                                           ;FUNCTION WHICH WILL BE SET
2481                                                                           ;TO FORCE THE ERROR.
2482  011670  000401                          BR      MF1
2483
2484           011672                         LOC=.                    ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
2485           011670                         LOC=-4&LOC
2486           011674                         LOC=LOC+4
2487           011674                         .=LOC
2488
2489  011674  000240                  MF1:    NOP
2490  011676  010412                          MOV     R4,(R2)          ;SET THE MAINTENANCE REGISTER.
```

```
2491  011700  021502                          CMP     (R5),R2         ;THIS REFERENCE TO (R5) WILL CAUSE A
2492  011702  005012                          CLR     (R2)            ;PARITY TRAP SINCE THE OTHER IN THAT
2493                                                                  ;PAIR WILL CAUSE A PARITY ERROR.
2494  011704  005701                          TST     R1              ;THIS WORD WILL CAUSE THE ERROR
2495  011706  000240                  MF2:    NOP                     ;WHEN THIS WORD IS REFERENCED.
2496
2497  011710                          MF3:                            ;REPORT ERROR.  MAINTENANCE
2498  011710  010437  001336                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2499                                                                  ;CAUSE ERROR.
2500  011714  104127                  1$:     ERROR   127
2501  011716  012737  177777  031102  MOV     #-1,MANFL2
2502  011724  000500                          BR      MFDONE
2503
2504  011726  022737  004404  177744  MFERRO: CMP     #4404,@#MEMERR  ;DID THE ERROR REGISTER
2505  011734  001042                          BNE     69$             ;SET PROPERLY?
2506
2507  011736  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2508  011740  005037  177572          65$:    CLR     @#MMR0
2509  011744  005037  172516                  CLR     @#MMR3
2510  011750  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2511  011756  005737  177744                  TST     @#MEMERR        ;REGISTER.
2512  011762  001416                          BEQ     68$
2513
2514  011764                          66$:                            ;ERROR REGISTER WON'T
2515  011764  013737  177740  001236          MOV     @#LOADRS,$TMP2  ;CLEAR
2516  011772  013737  177742  001240          MOV     @#HIADRS,$TMP3
2517  012000  013737  177744  001242          MOV     @#MEMERR,$TMP4
2518
2519  012006  104130                  67$:    ERROR   130
2520  012010  012737  177777  031062          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2521  012016  000443                          BR      MFDONE
2522
2523  012020  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2524  012026  001356                          BNE     66$             ;UNLOCKED.
2525  012030  022737  000003  177742          CMP     #3,@#HIADRS
2526  012036  001352                          BNE     66$
2527  012040  000432                          BR      MFDONE
2528
2529  012042                          69$:                            ;REPORT ERROR REGISTER
2530  012042  012637  001236                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2531  012046  005726                          TST     (SP)+           ;RESET THE STACK.
2532  012050  013737  177740  001240          MOV     @#LOADRS,$TMP3
2533  012056  013737  177742  001242          MOV     @#HIADRS,$TMP4
2534  012064  012737  010000  001244          MOV     #10000,$TMP5
2535  012072  012737  004404  001246          MOV     #4404,$TMP6
2536  012100  013737  177744  001250          MOV     @#MEMERR,$TMP7
2537
2538  012106  104131                  70$:    ERROR   131
2539  012110  012737  177777  031102          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2540  012116  012737  177777  031076          MOV     #-1,MMRFL2
2541  012124  000705                          BR      65$
2542  012126  104410                  MFDONE: RSET
2543
2544                                  ;;*****************************************************************
2545                                  ;*TEST 23        CACHE MAINTENANCE AND ERROR REGISTERS TEST 7
2546                                  ;*
```

```
                                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY TO FORCE
                                        ;*A PARITY ERROR ON THE MAIN MEMORY ODD WORD'S LOW BYTE,
                                        ;*WHEN THAT WORD IS THE UNWANTED WORD IN THE PAIR GOTTEN FROM MEMORY.
                                        ;*
                                        ;;*****************************************************************
012130  000004                        TST23:  SCOPE
012132  012737  000040  001302                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000023                  MG=$TN-1
                                                                        ;SET THE SKAD REGISTER
012140  012737  012450  030646                MOV     #TST24,SKAD     ;IN CASE THE TEST ABORTS.

012146  113737  001102  001232                MOVB    $TSTNM,$TMP0

012154  104415                               SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
012156  104416                               SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
012160  104417                               SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
012162  104420                               SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
012164  012704  040000                       MOV     #40000,R4       ;THIS PATTERN WILL BE PUT IN THE
012170  012704  177750                       MOV     #MAINT,R4       ;MAINTENANCE REGISTER.
012174  012737  012246  000114               MOV     #MGERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
012202  012737  000014  177746               MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
012210  000401                               BR      MG1

        012212                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        012210                          LOC=-4&LOC
        012214                          LOC=LOC+4
        012214                          .=LOC

012214  000240                  MG1:    NOP
012216  010412                          MOV     R4,(R2)         ;SET THE MAINTENANCE REGISTER.
012220  000240                          NOP                     ;THE REFERENCE TO THIS NOP
012222  005701                  MG2:    TST     R1              ;SHOULD CAUSE A PARITY ERROR TO OCCUR AT
                                                                ;MG2, RESULTING IN A TRAP!
012224  005012                          CLR     (R2)
012226  000240                          NOP

012230                          MG3:                            ;REPORT ERROR.  MAINTENANCE
012230  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
012234  104127                  1$:     ERROR   127
012236  012737  177777  031102  MOV     #-1,MANFL2
012244  000500                          BR      MGDONE

012246  022737  004410  177744  MGERR0: CMP     #4410,@#MEMERR  ;DID THE ERROR REGISTER
012254  001042                          BNE     69$             ;SET PROPERLY?

012256  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
012260  005037  177572          65$:    CLR     @#MMR0
012264  005037  172516                  CLR     @#MMR3
012270  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
012276  005737  177744                  TST     @#MEMERR        ;REGISTER.
012302  001416                          BEQ     68$

012304                          66$:                            ;ERROR REGISTER WON'T
012304  013737  177740  001236          MOV     @#LOADRS,$TMP2  ;CLEAR
012312  013737  177742  001240          MOV     @#HIADRS,$TMP3
```

```
2603  012320  013737  177744  001242          MOV     @#MEMERR,$TMP4
2604
2605  012326  104130                  67$:    ERROR   130
2606  012330  012737  177777  031062          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2607  012336  000443                          BR      MGDONE
2608
2609  012340  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2610  012346  001356                          BNE     66$             ;UNLOCKED.
2611  012350  022737  000003  177742          CMP     #3,@#HIADRS
2612  012356  001352                          BNE     66$
2613  012360  000432                          BR      MGDONE
2614
2615  012362                          69$:                            ;REPORT ERROR REGISTER
2616  012362  012637  001236                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2617  012366  005726                          TST     (SP)+           ;RESET THE STACK.
2618  012370  013737  177740  001240          MOV     @#LOADRS,$TMP3
2619  012376  013737  177742  001242          MOV     @#HIADRS,$TMP4
2620  012404  012737  040000  001244          MOV     #40000,$TMP5
2621  012412  012737  004410  001246          MOV     #4410,$TMP6
2622  012420  013737  177744  001250          MOV     @#MEMERR,$TMP7
2623
2624  012426  104131                  70$:    ERROR   131
2625  012430  012737  177777  031102          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2626  012436  012737  177777  031076          MOV     #-1,MMRFL2
2627  012444  000705                          BR      65$
2628  012446  104410                  MGDONE: RSET
2629
2630
2631                                  ;;**********************************************************************
2632                                  ;*TEST 24       CACHE MAINTENANCE AND ERROR REGISTERS TEST 10
2633                                  ;*
2634                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
2635                                  ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO, FOR THE
2636                                  ;*LOW BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
2637                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
2638                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
2639                                  ;*TO THE CACHE.
2640                                  ;*
2641                                  ;**********************************************************************
2642  012450  000004                  TST24:  SCOPE
2643  012452  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2644          000024          MH=$TN-1
2645                                                                  ;SET THE SKAD REGISTER
2646  012460  012737  013014  030646          MOV     #TST25,SKAD     ;IN CASE THE TEST ABORTS.
2647
2648  012466  113737  001102  001232          MOVB    $TSTNM,$TMP0
2649
2650  012474  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2651  012476  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2652  012500  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2653  012502  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2654  012504  012737  012612  000114          MOV     #MHERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
2655  012512  012704  000400                  MOV     #400,R4 ;PATTERN TO BE PUT IN MAINT. REG.
2656  012516  012702  177750                  MOV     #MAINT,R2
2657  012522  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
2658                                                                  ;FORCE MISS THE OTHER
```

```
2659
2660  012530  012705  012572                MOV     #MH1,R5         ;GROUP
2661  012534  005715                        TST     (R5)            ;MAKE MH1 A HIT IN
2662  012536  005715                        TST     (R5)            ;GROUP GP.
2663
2664                                                                ;SEE IF REFERENCE ADDRESS
2665  012540  032737  000010  177752        BIT     #10,@#HITMIS    ;IS A HIT.
2666  012546  001007                        BNE     1$
2667                                                                ;IF NOT ERROR!
2668  012550  010537  001236                MOV     R5,$TMP2
2669  012554  012737  000000  001234        MOV     #0,$TMP1
2670  012562  104001                        ERROR   1
2671
2672  012564  104411                        SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
2673
2674  012566  000240            1$:         NOP                     ;PUT THE PATTERN IN THE
2675  012570  010412                        MOV     R4,(R2)         ;MAINTENANCE REGISTER.
2676  012572  005012            MH1:        CLR     (R2)            ;THE FETCH OF THIS NEXT
2677                                                                ;INSTRUCTION SHOULD CAUSE
2678                                                                ;A PARITY ERROR IN THE
2679                                                                ;CACHE ADDRESS MEMORY GROUP GP.
2680
2681
2682  012574            MH2:                                        ;REPORT ERROR.  MAINTENANCE
2683  012574  010437  001236                MOV     R4,$TMP2        ;FUNCTION FAILED TO
2684                                                                ;CAUSE ERROR.
2685  012600  104127            1$:         ERROR   127
2686  012602  012737  177777  031102  MOV   #-1,MANFL2
2687  012610  000500                        BR      MHDONE
2688
2689  012612  022737  004420  177744  MHERR0: CMP   #4420,@#MEMERR  ;DID THE ERROR REGISTER
2690  012620  001042                        BNE     69$             ;SET PROPERLY?
2691
2692  012622  022626            64$:        CMP     (SP)+,(SP)+     ;RESET THE STACK
2693  012624  005037  177572    65$:        CLR     @#MMR0
2694  012630  005037  172516                CLR     @#MMR3
2695  012634  012737  177777  177744        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2696  012642  005737  177744                TST     @#MEMERR        ;REGISTER.
2697  012646  001416                        BEQ     68$
2698
2699  012650            66$:                                        ;ERROR REGISTER WON'T
2700  012650  013737  177740  001236        MOV     @#LOADRS,$TMP2  ;CLEAR
2701  012656  013737  177742  001240        MOV     @#HIADRS,$TMP3
2702  012664  013737  177744  001242        MOV     @#MEMERR,$TMP4
2703
2704  012672  104130            67$:        ERROR   130
2705  012674  012737  177777  031062        MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2706  012702  000443                        BR      MHDONE
2707
2708  012704  022737  177740  177740  68$:    CMP   #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2709  012712  001356                        BNE     66$             ;UNLOCKED.
2710  012714  022737  000003  177742        CMP     #3,@#HIADRS
2711  012722  001352                        BNE     66$
2712  012724  000432                        BR      MHDONE
2713
2714  012726            69$:                                        ;REPORT ERROR REGISTER
```

```
2715  012726  012637  001236                    MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2716  012732  005726                            TST     (SP)+           ;RESET THE STACK.
2717  012734  013737  177740  001240            MOV     @#LOADRS,$TMP3
2718  012742  013737  177742  001242            MOV     @#HIADRS,$TMP4
2719  012750  012737  000400  001244            MOV     #400,$TMP5
2720  012756  012737  004420  001246            MOV     #4420,$TMP6
2721  012764  013737  177744  001250            MOV     @#MEMERR,$TMP7
2722
2723  012772  104131                    70$:    ERROR   131
2724  012774  012737  177777  031102            MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2725  013002  012737  177777  031076            MOV     #-1,MMRFL2
2726  013010  000705                            BR      65$
2727  013012  104410                    MHDONE: RSET
2728
2729
2730                                    ;;**********************************************************************
2731                                    ;*TEST 25       CACHE MAINTENANCE AND ERROR REGISTERS TEST 11
2732                                    ;*
2733                                    ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
2734                                    ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ZERO, FOR THE
2735                                    ;*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
2736                                    ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
2737                                    ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
2738                                    ;*TO THE CACHE.
2739                                    ;*
2740                                    ;;**********************************************************************
2741  013014  000004                    TST25:  SCOPE
2742  013016  012737  000040  001302            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2743            000025                  MI=$TN-1
2744                                                                    ;SET THE SKAD REGISTER
2745  013024  012737  013360  030646            MOV     #TST26,SKAD     ;IN CASE THE TEST ABORTS.
2746
2747  013032  113737  001102  001232            MOVB    $TSTNM,$TMPO
2748
2749  013040  104415                            SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
2750  013042  104416                            SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
2751  013044  104417                            SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
2752  013046  104420                            SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
2753  013050  012737  013156  000114            MOV     #MIERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
2754  013056  012704  001000                    MOV     #1000,R4        ;PATTERN TO BE PUT IN MAINT. REG.
2755  013062  012702  177750                    MOV     #MAINT,R2
2756  013066  012737  000030  177746            MOV     #SDM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
2757                                                                    ;FORCE MISS THE OTHER
2758                                                                    ;GROUP
2759  013074  012705  013136                    MOV     #MI1,R5         ;MAKE MI1 A HIT IN
2760  013100  005715                            TST     (R5)            ;GROUP GP.
2761  013102  005715                            TST     (R5)
2762
2763                                                                    ;SEE IF REFERENCE ADDRESS
2764  013104  032737  000010  177752            BIT     #10,@#HITMIS    ;IS A HIT.
2765  013112  001007                            BNE     1$
2766                                                                    ;IF NOT ERROR!
2767  013114  010537  001236                    MOV     R5,$TMP2
2768  013120  012737  000000  001234            MOV     #0,$TMP1
2769  013126  104001                            ERROR   1
2770
```

```
2771  013130  104411                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
2772
2773  013132  000240              1$:         NOP                     ;PUT THE PATTERN IN THE
2774  013134  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
2775  013136  005012              MI1:        CLR     (R2)            ;THE FETCH OF THIS NEXT
2776                                                                  ;INSTRUCTION SHOULD CAUSE
2777                                                                  ;A PARITY ERROR IN THE
2778                                                                  ;CACHE ADDRESS MEMORY GROUP GP.
2779
2780
2781  013140                      MI2:                                ;REPORT ERROR.  MAINTENANCE
2782  013140  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
2783                                                                  ;CAUSE ERROR.
2784  013144  104127              1$:         ERROR   127
2785  013146  012737  177777  031102  MOV     #-1,MANFL2
2786  013154  000500                          BR      MIDONE
2787
2788  013156  022737  004420  177744  MIERRO: CMP     #4420,@#MEMERR  ;DID THE ERROR REGISTER
2789  013164  001042                          BNE     69$             ;SET PROPERLY?
2790
2791  013166  022626              64$:        CMP     (SP)+,(SP)+     ;RESET THE STACK
2792  013170  005037  177572      65$:        CLR     @#MMR0
2793  013174  005037  172516                  CLR     @#MMR3
2794  013200  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2795  013206  005737  177744                  TST     @#MEMERR        ;REGISTER.
2796  013212  001416                          BEQ     68$
2797
2798  013214                      66$:                                ;ERROR REGISTER WON'T
2799  013214  013737  177740  001236          MOV     @#LOADRS,$TMP2  ;CLEAR
2800  013222  013737  177742  001240          MOV     @#HIADRS,$TMP3
2801  013230  013737  177744  001242          MOV     @#MEMERR,$TMP4
2802
2803  013236  104130              67$:        ERROR   130
2804  013240  012737  177777  031062          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2805  013246  000443                          BR      MIDONE
2806
2807  013250  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2808  013256  001356                          BNE     66$             ;UNLOCKED.
2809  013260  022737  000003  177742          CMP     #3,@#HIADRS
2810  013266  001352                          BNE     66$
2811  013270  000432                          BR      MIDONE
2812
2813  013272                      69$:                                ;REPORT ERROR REGISTER
2814  013272  012637  001236                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2815  013276  005726                          TST     (SP)+           ;RESET THE STACK.
2816  013300  013737  177740  001240          MOV     @#LOADRS,$TMP3
2817  013306  013737  177742  001242          MOV     @#HIADRS,$TMP4
2818  013314  012737  001000  001244          MOV     #1000,$TMP5
2819  013322  012737  004420  001246          MOV     #4420,$TMP6
2820  013330  013737  177744  001250          MOV     @#MEMERR,$TMP7
2821
2822  013336  104131              70$:        ERROR   131
2823  013340  012737  177777  031102          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2824  013346  012737  177777  031076          MOV     #-1,MMRFL2
2825  013354  000705                          BR      65$
2826  013356  104410              MIDONE:     RSET
```

```
                                        ;;*****************************************************************
                                        ;*TEST 26       CACHE MAINTENANCE AND ERROR REGISTERS TEST 12
                                        ;*
                                        ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
                                        ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE
                                        ;*LOW BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
                                        ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
                                        ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
                                        ;*TO THE CACHE.
                                        ;*
                                        ;;*****************************************************************
013360  000004                          TST26:  SCOPE
013362  012737  000040  001302                  MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000026                  MJ=$TN-1
                                                                        ;SET THE SKAD REGISTER
013370  012737  013724  030646          MOV     #TST27,SKAD             ;IN CASE THE TEST ABORTS.

013376  113737  001102  001232          MOVB    $TSTNM,$TMPO

013404  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
013406  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
013410  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
013412  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
013414  012737  013522  000114          MOV     #MJERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
013422  012704  002000                  MOV     #2000,R4                ;PATTERN TO BE PUT IN MAINT. REG.
013426  012702  177750                  MOV     #MAINT,R2
013432  012737  000044  177746          MOV     #S1MD,@#CONTRL  ;FORCE SELECT GROUP 1 AND
                                                                ;FORCE MISS THE OTHER
                                                                ;GROUP
013440  012705  013502                  MOV     #MJ1,R5         ;MAKE MJ1 A HIT IN
013444  005715                          TST     (R5)            ;GROUP GP.
013446  005715                          TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
013450  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
013456  001007                          BNE     1$

                                                                ;IF NOT ERROR!
013460  010537  001236                  MOV     R5,$TMP2
013464  012737  000001  001234          MOV     #1,$TMP1
013472  104001                          ERROR   1

013474  104411                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

013476  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
013500  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
013502  005012                  MJ1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
                                                                ;CACHE ADDRESS MEMORY GROUP GP.

013504                          MJ2:                            ;REPORT ERROR.  MAINTENANCE
013504  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
```

```
2883  013510  104127                    1$:     ERROR   127
2884  013512  012737  177777  031102    MOV     #-1,MANFL2
2885  013520  000500                    BR      MJDONE
2886
2887  013522  022737  004440  177744    MJERRO: CMP     #4440,@#MEMERR  ;DID THE ERROR REGISTER
2888  013530  001042                    BNE     69$             ;SET PROPERLY?
2889
2890  013532  022626                    64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
2891  013534  005037  177572            65$:    CLR     @#MMR0
2892  013540  005037  172516            CLR     @#MMR3
2893  013544  012737  177777  177744    MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
2894  013552  005737  177744            TST     @#MEMERR        ;REGISTER.
2895  013556  001416                    BEQ     68$
2896
2897  013560                            66$:                    ;ERROR REGISTER WON'T
2898  013560  013737  177740  001236    MOV     @#LOADRS,$TMP2  ;CLEAR
2899  013566  013737  177742  001240    MOV     @#HIADRS,$TMP3
2900  013574  013737  177744  001242    MOV     @#MEMERR,$TMP4
2901
2902  013602  104130                    67$:    ERROR   130
2903  013604  012737  177777  031062    MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
2904  013612  000443                    BR      MJDONE
2905
2906  013614  022737  177740  177740    68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
2907  013622  001356                    BNE     66$             ;UNLOCKED.
2908  013624  022737  000003  177742    CMP     #3,@#HIADRS
2909  013632  001352                    BNE     66$
2910  013634  000432                    BR      MJDONE
2911
2912  013636                            69$:                    ;REPORT ERROR REGISTER
2913  013636  012637  001236            MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
2914  013642  005726                    TST     (SP)+           ;RESET THE STACK.
2915  013644  013737  177740  001240    MOV     @#LOADRS,$TMP3
2916  013652  013737  177742  001242    MOV     @#HIADRS,$TMP4
2917  013660  012737  002000  001244    MOV     #2000,$TMP5
2918  013666  012737  004440  001246    MOV     #4440,$TMP6
2919  013674  013737  177744  001250    MOV     @#MEMERR,$TMP7
2920
2921  013702  104131                    70$:    ERROR   131
2922  013704  012737  177777  031102    MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
2923  013712  012737  177777  031076    MOV     #-1,MMRFL2
2924  013720  000705                    BR      65$
2925  013722  104410                    MJDONE: RSET
2926
2927
2928                                    ;;*****************************************************************
2929                                    ;*TEST 27       CACHE MAINTENANCE AND ERROR REGISTERS TEST 13
2930                                    ;*
2931                                    ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
2932                                    ;*TO FORCE A PARITY ERROR IN THE CACHE ADDRESS MEMORY OF GROUP ONE, FOR THE
2933                                    ;*HIGH BYTE OF THE ADDRESS WORD. ALSO TESTED IS THE ERROR REGISTER'S
2934                                    ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
2935                                    ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
2936                                    ;*TO THE CACHE.
2937                                    ;*
2938                                    ;;*****************************************************************
```

```
                                TST27:  SCOPE
013724  000004
013726  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000027                  MK=$TN-1
                                                                ;SET THE SKAD REGISTER
013734  012737  014270  030646          MOV     #TST30,SKAD     ;IN CASE THE TEST ABORTS.

013742  113737  001102  001232          MOVB    $TSTNM,$TMP0

013750  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
013752  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
013754  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
013756  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
013760  012737  014066  000114          MOV     #MKERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
013766  012704  004000                  MOV     #4000,R4        ;PATTERN TO BE PUT IN MAINT. REG.
013772  012702  177750                  MOV     #MAINT,R2
013776  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
                                                                ;FORCE MISS THE OTHER
                                                                ;GROUP
014004  012705  014046                  MOV     #MK1,R5         ;MAKE MK1 A HIT IN
014010  005715                          TST     (R5)            ;GROUP GP.
014012  005715                          TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
014014  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
014022  001007                          BNE     1$

                                                                ;IF NOT ERROR!
014024  010537  001236                  MOV     R5,$TMP2
014030  012737  000001  001234          MOV     #1,$TMP1
014036  104001                          ERROR   1

014040  104411                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

014042  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
014044  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
014046  005012                  MK1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
                                                                ;CACHE ADDRESS MEMORY GROUP GP.

014050                          MK2:                            ;REPORT ERROR.  MAINTENANCE
014050  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
014054  104127                  1$:     ERROR   127
014056  012737  177777  031102  MOV     #-1,MANFL2
014064  000500                          BR      MKDONE

014066  022737  004440  177744  MKERR0: CMP     #4440,@#MEMERR  ;DID THE ERROR REGISTER
014074  001042                          BNE     69$             ;SET PROPERLY?

014076  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
014100  005037  177572          65$:    CLR     @#MMR0
014104  005037  172516                  CLR     @#MMR3
014110  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
014116  005737  177744                  TST     @#MEMERR        ;REGISTER.
014122  001416                          BEQ     68$
```

```
2995
2996  014124                                  66$:                                    ;ERROR REGISTER WON'T
2997  014124  013737  177740  001236                MOV     @#LOADRS,$TMP2          ;CLEAR
2998  014132  013737  177742  001240                MOV     @#HIADRS,$TMP3
2999  014140  013737  177744  001242                MOV     @#MEMERR,$TMP4
3000
3001  014146  104130                          67$:  ERROR   130
3002  014150  012737  177777  031062                MOV     #-1,MMRFLG              ;SIGNAL BAD REGISTER
3003  014156  000443                                BR      MKDONE
3004
3005  014160  022737  177740  177740          68$:  CMP     #177740,@#LOADRS        ;SEE IF ADDRESS REGISTER
3006  014166  001356                                BNE     66$                     ;UNLOCKED.
3007  014170  022737  000003  177742                CMP     #3,@#HIADRS
3008  014176  001352                                BNE     66$
3009  014200  000432                                BR      MKDONE
3010
3011  014202                                  69$:                                    ;REPORT ERROR REGISTER
3012  014202  012637  001236                      MOV     (SP)+,$TMP2             ;NOT SET AS EXPECTED.
3013  014206  005726                                TST     (SP)+                   ;RESET THE STACK.
3014  014210  013737  177740  001240                MOV     @#LOADRS,$TMP3
3015  014216  013737  177742  001242                MOV     @#HIADRS,$TMP4
3016  014224  012737  004000  001244                MOV     #4000,$TMP5
3017  014232  012737  004440  001246                MOV     #4440,$TMP6
3018  014240  013737  177744  001250                MOV     @#MEMERR,$TMP7
3019
3020  014246  104131                          70$:  ERROR   131
3021  014250  012737  177777  031102                MOV     #-1,MANFL2              ;SIGNAL BAD REGISTER
3022  014256  012737  177777  031076                MOV     #-1,MMRFL2
3023  014264  000705                                BR      65$
3024  014266  104410                          MKDONE: RSET
3025
3026
3027                                          ;;*****************************************************************
3028                                          ;*TEST 30       CACHE MAINTENANCE AND ERROR REGISTERS TEST 14
3029                                          ;*
3030                                          ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3031                                          ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ZERO, FOR THE
3032                                          ;*LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3033                                          ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3034                                          ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3035                                          ;*TO THE CACHE.
3036                                          ;*
3037                                          ;;*****************************************************************
3038  014270  000004                          TST30:  SCOPE
3039  014272  012737  000040  001302                MOV     #40,$TIMES              ;;DO 40 ITERATIONS
3040          000030                          ML=$TN-1
3041                                                                                  ;SET THE SKAD REGISTER
3042  014300  012737  014534  030646                MOV     #TST31,SKAD             ;IN CASE THE TEST ABORTS.
3043
3044  014306  113737  001102  001232                MOVB    $TSTNM,$TMP0
3045
3046  014314  104415                                SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3047  014316  104416                                SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3048  014320  104417                                SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3049  014322  104420                                SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3050  014324  012737  014432  000114                MOV     #MLERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
```

```
3051  014332  012704  000020                  MOV     #20,R4  ;PATTERN TO BE PUT IN MAINT. REG.
3052  014336  012702  177750                  MOV     #MAINT,R2
3053  014342  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
3054                                                                  ;FORCE MISS THE OTHER
3055                                                                  ;GROUP
3056  014350  012705  014412                  MOV     #ML1,R5         ;MAKE ML1 A HIT IN
3057  014354  005715                          TST     (R5)            ;GROUP GP.
3058  014356  005715                          TST     (R5)
3059
3060                                                                  ;SEE IF REFERENCE ADDRESS
3061  014360  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
3062  014366  001007                          BNE     1$
3063                                                                  ;IF NOT ERROR!
3064  014370  010537  001236                  MOV     R5,$TMP2
3065  014374  012737  000000  001234          MOV     #0,$TMP1
3066  014402  104001                          ERROR   1
3067
3068  014404  104411                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
3069
3070  014406  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
3071  014410  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
3072  014412  005012                  ML1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
3073                                                                  ;INSTRUCTION SHOULD CAUSE
3074                                                                  ;A PARITY ERROR IN THE
3075                                                                  ;CACHE DATA MEMORY GROUP GP.
3076
3077
3078  014414                          ML2:                            ;REPORT ERROR.  MAINTENANCE
3079  014414  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
3080                                                                  ;CAUSE ERROR.
3081  014420  104127                  1$:     ERROR   127
3082  014422  012737  177777  031102  MOV     #-1,MANFL2
3083  014430  000500                          BR      MLDONE
3084
3085  014432  022737  004500  177744  MLERR0: CMP     #4500,@#MEMERR  ;DID THE ERROR REGISTER
3086  014440  001042                          BNE     69$             ;SET PROPERLY?
3087
3088  014442  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3089  014444  005037  177572          65$:    CLR     @#MMR0
3090  014450  005037  172516                  CLR     @#MMR3
3091  014454  012737  177777  177744          MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3092  014462  005737  177744                  TST     @#MEMERR        ;REGISTER.
3093  014466  001416                          BEQ     68$
3094
3095  014470                          66$:                            ;ERROR REGISTER WON'T
3096  014470  013737  177740  001236          MOV     @#LOADRS,$TMP2  ;CLEAR
3097  014476  013737  177742  001240          MOV     @#HIADRS,$TMP3
3098  014504  013737  177744  001242          MOV     @#MEMERR,$TMP4
3099
3100  014512  104130                  67$:    ERROR   130
3101  014514  012737  177777  031062          MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3102  014522  000443                          BR      MLDONE
3103
3104  014524  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3105  014532  001356                          BNE     66$             ;UNLOCKED.
3106  014534  022737  000003  177742          CMP     #3,@#HIADRS
```

```
3107  014542  001352                          BNE     66$
3108  014544  000432                          BR      MLDONE
3109
3110  014546                          69$:                            ;REPORT ERROR REGISTER
3111  014546  012637  001236                  MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3112  014552  005726                          TST     (SP)+           ;RESET THE STACK.
3113  014554  013737  177740  001240          MOV     @#LOADRS,$TMP3
3114  014562  013737  177742  001242          MOV     @#HIADRS,$TMP4
3115  014570  012737  000020  001244          MOV     #20,$TMP5
3116  014576  012737  004500  001246          MOV     #4500,$TMP6
3117  014604  013737  177744  001250          MOV     @#MEMERR,$TMP7
3118
3119  014612  104131                  70$:    ERROR   131
3120  014614  012737  177777  031102          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3121  014622  012737  177777  031076          MOV     #-1,MMRFL2
3122  014630  000705                          BR      65$
3123  014632  104410                  MLDONE: RSET
3124
3125
3126                                  ;;*****************************************************************
3127                                  ;*TEST 31        CACHE MAINTENANCE AND ERROR REGISTERS TEST 15
3128                                  ;*
3129                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3130                                  ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ZERO, FOR THE
3131                                  ;*HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3132                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3133                                  ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3134                                  ;*TO THE CACHE.
3135                                  ;*
3136                                  ;;*****************************************************************
3137  014634  000004                  TST31:  SCOPE
3138  014636  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3139          000031                  MN=$TN-1
3140                                                                  ;SET THE SKAD REGISTER
3141  014644  012737  015200  030646          MOV     #TST32,SKAD     ;IN CASE THE TEST ABORTS.
3142
3143  014652  113737  001102  001232          MOVB    $TSTNM,$TMP0
3144
3145  014660  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3146  014662  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3147  014664  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3148  014666  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3149  014670  012737  014776  000114          MOV     #NMERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
3150  014676  012704  000040                  MOV     #40,R4  ;PATTERN TO BE PUT IN MAINT. REG.
3151  014702  012702  177750                  MOV     #MAINT,R2
3152  014706  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;FORCE SELECT GROUP 0 AND
3153                                                                  ;FORCE MISS THE OTHER
3154                                                                  ;GROUP
3155  014714  012705  014756                  MOV     #NM1,R5         ;MAKE NM1 A HIT IN
3156  014720  005715                          TST     (R5)            ;GROUP GP.
3157  014722  005715                          TST     (R5)
3158
3159                                                                  ;SEE IF REFERENCE ADDRESS
3160  014724  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
3161  014732  001007                          BNE     1$
3162                                                                  ;IF NOT ERROR!
```

```
3163  014734  010537  001236          MOV     R5,$TMP2
3164  014740  012737  000000  001234  MOV     #0,$TMP1
3165  014746  104001                  ERROR   1
3166
3167  014750  104411                  SKIPT                       ;ERROR FATAL. GO TO NEXT TEST.
3168
3169  014752  000240          1$:     NOP                         ;PUT THE PATTERN IN THE
3170  014754  010412                  MOV     R4,(R2)             ;MAINTENANCE REGISTER.
3171  014756  005012          NM1:    CLR     (R2)                ;THE FETCH OF THIS NEXT
3172                                                              ;INSTRUCTION SHOULD CAUSE
3173                                                              ;A PARITY ERROR IN THE
3174                                                              ;CACHE DATA MEMORY GROUP GP.
3175
3176
3177  014760          NM2:                                        ;REPORT ERROR.  MAINTENANCE
3178  014760  010437  001236          MOV     R4,$TMP2            ;FUNCTION FAILED TO
3179                                                              ;CAUSE ERROR.
3180  014764  104127          1$:     ERROR   127
3181  014766  012737  177777  031102  MOV     #-1,MANFL2
3182  014774  000500                  BR      NMDONE
3183
3184  014776  022737  004500  177744  NMERRO: CMP     #4500,@#MEMERR      ;DID THE ERROR REGISTER
3185  015004  001042                  BNE     69$                 ;SET PROPERLY?
3186
3187  015006  022626          64$:    CMP     (SP)+,(SP)+         ;RESET THE STACK
3188  015010  005037  177572  65$:    CLR     @#MMR0
3189  015014  005037  172516          CLR     @#MMR3
3190  015020  012737  177777  177744  MOV     #-1,@#MEMERR        ;TRY TO CLEAR THE ERROR
3191  015026  005737  177744          TST     @#MEMERR            ;REGISTER.
3192  015032  001416                  BEQ     68$
3193
3194  015034          66$:                                        ;ERROR REGISTER WON'T
3195  015034  013737  177740  001236  MOV     @#LOADRS,$TMP2      ;CLEAR
3196  015042  013737  177742  001240  MOV     @#HIADRS,$TMP3
3197  015050  013737  177744  001242  MOV     @#MEMERR,$TMP4
3198
3199  015056  104130          67$:    ERROR   130
3200  015060  012737  177777  031062  MOV     #-1,MMRFLG          ;SIGNAL BAD REGISTER
3201  015066  000443                  BR      NMDONE
3202
3203  015070  022737  177740  177740  68$:    CMP     #177740,@#LOADRS    ;SEE IF ADDRESS REGISTER
3204  015076  001356                  BNE     66$                 ;UNLOCKED.
3205  015100  022737  000003  177742  CMP     #3,@#HIADRS
3206  015106  001352                  BNE     66$
3207  015110  000432                  BR      NMDONE
3208
3209  015112          69$:                                        ;REPORT ERROR REGISTER
3210  015112  012637  001236          MOV     (SP)+,$TMP2         ;NOT SET AS EXPECTED.
3211  015116  005726                  TST     (SP)+               ;RESET THE STACK.
3212  015120  013737  177740  001240  MOV     @#LOADRS,$TMP3
3213  015126  013737  177742  001242  MOV     @#HIADRS,$TMP4
3214  015134  012737  000040  001244  MOV     #40,$TMP5
3215  015142  012737  004500  001246  MOV     #4500,$TMP6
3216  015150  013737  177744  001250  MOV     @#MEMERR,$TMP7
3217
3218  015156  104131          70$:    ERROR   131
```

```
3219  015160  012737  177777  031102          MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3220  015166  012737  177777  031076          MOV     #-1,MMRFL2
3221  015174  000705                          BR      65$
3222  015176  104410                  NMDONE: RSET
3223
3224
3225                                          ;;*****************************************************************
3226                                          ;*TEST 32       CACHE MAINTENANCE AND ERROR REGISTERS TEST 16
3227                                          ;*
3228                                          ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3229                                          ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
3230                                          ;*LOW BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3231                                          ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
3232                                          ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
3233                                          ;*TO THE CACHE.
3234                                          ;*
3235                                          ;*****************************************************************
3236  015200  000004                  TST32:  SCOPE
3237  015202  012737  000040  001302          MOV     #40.,$TIMES     ;,DO 40 ITERATIONS
3238           000032                 M0=$TN-1
3239                                                                  ;SET THE SKAD REGISTER
3240  015210  012737  015544  030646          MOV     #TST33,SKAD     ;IN CASE THE TEST ABORTS.
3241
3242  015216  113737  001102  001232          MOVB    $TSTNM,$TMP0
3243
3244  015224  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3245  015226  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3246  015230  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3247  015232  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3248  015234  012737  015342  000114          MOV     #M0ERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
3249  015242  012704  000100                  MOV     #100,R4 ;PATTERN TO BE PUT IN MAINT. REG.
3250  015246  012702  177750                  MOV     #MAINT,R2
3251  015252  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
3252                                                                  ;FORCE MISS THE OTHER
3253                                                                  ;GROUP
3254  015260  012705  015322                  MOV     #M01,R5         ;MAKE M01 A HIT IN
3255  015264  005715                          TST     (R5)            ;GROUP GP.
3256  015266  005715                          TST     (R5)
3257
3258                                                                  ;SEE IF REFERENCE ADDRESS
3259  015270  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
3260  015276  001007                          BNE     1$
3261                                                                  ;IF NOT ERROR!
3262  015300  010537  001236                  MOV     R5,$TMP2
3263  015304  012737  000001  001234          MOV     #1,$TMP1
3264  015312  104001                          ERROR   1
3265
3266  015314  104411                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.
3267
3268  015316  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
3269  015320  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
3270  015322  005012                  M01:    CLR     (R2)            ;THE FETCH OF THIS NEXT
3271                                                                  ;INSTRUCTION SHOULD CAUSE
3272                                                                  ;A PARITY ERROR IN THE
3273                                                                  ;CACHE DATA MEMORY GROUP GP.
3274
```

```
3275
3276  015324                          MO2:                                ;REPORT ERROR.  MAINTENANCE
3277  015324  010437  001236                  MOV     R4,$TMP2            ;FUNCTION FAILED TO
3278                                                                      ;CAUSE ERROR.
3279  015330  104127                  1$:     ERROR   127
3280  015332  012737  177777  031102  MOV     #-1,MANFL2
3281  015340  000500                  BR      MODONE
3282
3283  015342  022737  004600  177744  MOERRO: CMP     #4600,@#MEMERR      ;DID THE ERROR REGISTER
3284  015350  001042                  BNE     69$                         ;SET PROPERLY?
3285
3286  015352  022626                  64$:    CMP     (SP)+,(SP)+         ;RESET THE STACK
3287  015354  005037  177572          65$:    CLR     @#MMR0
3288  015360  005037  172516                  CLR     @#MMR3
3289  015364  012737  177777  177744          MOV     #-1,@#MEMERR        ;TRY TO CLEAR THE ERROR
3290  015372  005737  177744                  TST     @#MEMERR            ;REGISTER.
3291  015376  001416                  BEQ     68$
3292
3293  015400                          66$:                                ;ERROR REGISTER WON'T
3294  015400  013737  177740  001236          MOV     @#LOADRS,$TMP2      ;CLEAR
3295  015406  013737  177742  001240          MOV     @#HIADRS,$TMP3
3296  015414  013737  177744  001242          MOV     @#MEMERR,$TMP4
3297
3298  015422  104130                  67$:    ERROR   130
3299  015424  012737  177777  031062          MOV     #-1,MMRFLG          ;SIGNAL BAD REGISTER
3300  015432  000443                  BR      MODONE
3301
3302  015434  022737  177740  177740  68$:    CMP     #177740,@#LOADRS    ;SEE IF ADDRESS REGISTER
3303  015442  001356                  BNE     66$                         ;UNLOCKED.
3304  015444  022737  000003  177742          CMP     #3,@#HIADRS
3305  015452  001352                  BNE     66$
3306  015454  000432                  BR      MODONE
3307
3308  015456                          69$:                                ;REPORT ERROR REGISTER
3309  015456  012637  001236                  MOV     (SP)+,$TMP2         ;NOT SET AS EXPECTED.
3310  015462  005726                  TST     (SP)+                       ;RESET THE STACK.
3311  015464  013737  177740  001240          MOV     @#LOADRS,$TMP3
3312  015472  013737  177742  001242          MOV     @#HIADRS,$TMP4
3313  015500  012737  000100  001244          MOV     #100,$TMP5
3314  015506  012737  004600  001246          MOV     #4600,$TMP6
3315  015514  013737  177744  001250          MOV     @#MEMERR,$TMP7
3316
3317  015522  104131                  70$:    ERROR   131
3318  015524  012737  177777  031102          MOV     #-1,MANFL2          ;SIGNAL BAD REGISTER
3319  015532  012737  177777  031076          MOV     #-1,MMRFL2
3320  015540  000705                  BR      65$
3321  015542  104410                  MODONE: RSET
3322
3323
3324                                  ;;*****************************************************************
3325                                  ;*TEST 33       CACHE MAINTENANCE AND ERROR REGISTERS TEST 17
3326                                  ;*
3327                                  ;*THIS IS A TEST OF THE MAINTENANCE REGISTER'S ABILITY
3328                                  ;*TO FORCE A PARITY ERROR IN THE CACHE DATA MEMORY OF GROUP ONE, FOR THE
3329                                  ;*HIGH BYTE OF THE DATA WORD. ALSO TESTED IS THE ERROR REGISTER'S
3330                                  ;*ABILITY TO SET CORRECTLY FOR THIS ERROR.
```

```
                                        ;*THE REFERENCE RESULTING IN THIS ERROR IS MADE DIRECTLY FROM THE CPU
                                        ;*TO THE CACHE.
                                        ;*
                                        ;;*****************************************************************
015544  000004                  $TST33:  SCOPE
015546  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000033          MP=$TN-1
                                                                ;SET THE SKAD REGISTER
015554  012737  016110  030646          MOV     #TST34,@SKAD    ;IN CASE THE TEST ABORTS.

015562  113737  001102  001232          MOVB    $TSTNM,$TMP0

015570  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
015572  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
015574  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
015576  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
015600  012737  015706  000114          MOV     #MPERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
015606  012704  000200                  MOV     #200,R4 ;PATTERN TO BE PUT IN MAINT. REG.
015612  012702  177750                  MOV     #MAINT,R2
015616  012737  000044  177746          MOV     #S1M0,@#CONTRL  ;FORCE SELECT GROUP 1 AND
                                                                ;FORCE MISS THE OTHER
                                                                ;GROUP
015624  012705  015666                  MOV     #MP1,R5         ;MAKE MP1 A HIT IN
015630  005715                          TST     (R5)            ;GROUP GP.
015632  005715                          TST     (R5)

                                                                ;SEE IF REFERENCE ADDRESS
015634  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
015642  001007                          BNE     1$

                                                                ;IF NOT ERROR!
015644  010537  001236                  MOV     R5,$TMP2
015650  012737  000001  001234          MOV     #1,$TMP1
015656  104001                          ERROR   1

015660  104411                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

015662  000240                  1$:     NOP                     ;PUT THE PATTERN IN THE
015664  010412                          MOV     R4,(R2)         ;MAINTENANCE REGISTER.
015666  005012                  MP1:    CLR     (R2)            ;THE FETCH OF THIS NEXT
                                                                ;INSTRUCTION SHOULD CAUSE
                                                                ;A PARITY ERROR IN THE
                                                                ;CACHE DATA MEMORY GROUP GP.

015670                          MP2:                            ;REPORT ERROR.  MAINTENANCE
015670  010437  001236                  MOV     R4,$TMP2        ;FUNCTION FAILED TO
                                                                ;CAUSE ERROR.
015674  104127                  1$:     ERROR   127
015676  012737  177777  031102  MOV     #-1,MANFL2
015704  000500                          BR      MPDONE

015706  022737  004600  177744  MPERRO: CMP     #4600,@#MEMERR  ;DID THE ERROR REGISTER
015714  001042                          BNE     69$             ;SET PROPERLY?

015716  022626                  64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
015720  005037  177572          65$:    CLR     @#MMR0
```

```
3387  015724  005037  172516                    CLR     @#MMR3
3388  015730  012737  177777  177744            MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3389  015736  005737  177744                    TST     @#MEMERR        ;REGISTER.
3390  015742  001416                            BEQ     68$
3391
3392  015744                          66$:                            ;ERROR REGISTER WON'T
3393  015744  013737  177740  001236            MOV     @#LOADRS,$TMP2  ;CLEAR
3394  015752  013737  177742  001240            MOV     @#HIADRS,$TMP3
3395  015760  013737  177744  001242            MOV     @#MEMERR,$TMP4
3396
3397  015766  104130                  67$:      ERROR   130
3398  015770  012737  177777  031062            MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3399  015776  000443                            BR      MPDONE
3400
3401  016000  022737  177740  177740  68$:      CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3402  016006  001356                            BNE     66$             ;UNLOCKED.
3403  016010  022737  000003  177742            CMP     #3,@#HIADRS
3404  016016  001352                            BNE     66$
3405  016020  000432                            BR      MPDONE
3406
3407  016022                          69$:                            ;REPORT ERROR REGISTER
3408  016022  012637  001236                    MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3409  016026  005726                            TST     (SP)+           ;RESET THE STACK.
3410  016030  013737  177740  001240            MOV     @#LOADRS,$TMP3
3411  016036  013737  177742  001242            MOV     @#HIADRS,$TMP4
3412  016044  012737  000200  001244            MOV     #200,$TMP5
3413  016052  012737  004600  001246            MOV     #4600,$TMP6
3414  016060  013737  177744  001250            MOV     @#MEMERR,$TMP7
3415
3416  016066  104131                  70$:      ERROR   131
3417  016070  012737  177777  031102            MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3418  016076  012737  177777  031076            MOV     #-1,MMRFL2
3419  016104  000705                            BR      65$
3420  016106  104410                  MPDONE:   RSET
3421
3422
3423
3424
3425
3426                                  ;;**********************************************************************
3427                                  ;*TEST 34        CACHE MAINTENANCE AND ERROR REGISTERS TEST 20
3428                                  ;*
3429                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3430                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3431                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3432                                  ;*THE MAINTENANCE REGISTER IS USED TO MAKE THAT REFERENCE CAUSE A
3433                                  ;*MAIN MEMORY ADDRESS AND CONTROL LINES PARITY ERROR ON THE
3434                                  ;*MAIN MEMORY BUS.
3435                                  ;*
3436                                  ;;**********************************************************************
3437  016110  000004                  TST34:    SCOPE
3438  016112  012737  000040  001302            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3439          000034                  MR=$TN-1
3440                                                                    ;SET THE SKAD REGISTER
3441  016120  012737  016540  030646            MOV     #TST35,SKAD     ;IN CASE THE TEST ABORTS.
3442
```

```
3443  016126  113737  001102  001232          MOVB    $TSTNM,$TMPO
3444
3445  016134  104415                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3446  016136  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3447  016140  104417                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3448  016142  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3449  016144  104412                          MMSKIP
3450  016146  012737  016330  000114          MOV     #MRERRO,@#CACHVEC       ;SET UP FOR THE ERROR.
3451  016154  012737  000474  000004          MOV     #CPSPUR,@#ERRVEC        ;NOTE THAT WHEN THIS ERROR
3452                                                                          ;ON THE MAIN MEMORY ADDRESS
3453                                                                          ;AND CONTROL LINES OCCURS
3454                                                                          ;A TIME OUT WILL RESULT ON THE
3455                                                                          ;UNIBUS!! THIS WILL CAUSE A
3456                                                                          ;TRAP TO VECTOR ERRVEC BEFORE
3457                                                                          ;THE TRAP TO CACHVEC OCCURS! BOTH
3458                                                                          ;WILL OCCUR!
3459  016162  012746  177777                  MOV     #-1,-(SP)               ;PUT A MARKER ON THE STACK
3460
3461  016166  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
3462                                                                  ;TO RELOCATE EVERYTHING
3463  016172  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
3464  016176  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY.
3465  016202  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
3466  016204  012705  170200                  MOV     #MAPLOD,R5      ;THAT ADDRESS ARE
3467                                                                  ;RELOCATED TO THEMSELVES.
3468  016210  012722  077406          64$:    MOV     #77406,(R2)+
3469  016214  010401                          MOV     R4,R1
3470  016216  072127  000006                  ASH     #6,R1
3471  016222  010125                          MOV     R1,(R5)+
3472  016224  005025                          CLR     (R5)+
3473  016226  010410                          MOV     R4,(R0)
3474  016230  062720  170000                  ADD     #170000,(R0)+
3475  016234  062704  000200                  ADD     #200,R4
3476  016240  077315                          SOB     R3,64$
3477  016242  012710  177600                  MOV     #177600,(R0)
3478  016246  012712  077406                  MOV     #77406,(R2)
3479
3480  016252  012737  000060  172516          MOV     #60,@#MMR3              ;TURN ON THE MAPPING BOX AND
3481  016260  012737  000001  177572          MOV     #1,@#MMR0               ;ENABLE 22 BIT MODE ADDRESSING.
3482
3483  016266  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
3484  016274  012702  177750                  MOV     #MAINT,R2
3485  016300  000240                          NOP                             ;FOR SCOPING WITH AN OSCILLOSCOPE!
3486  016302  012712  000002                  MOV     #2,(R2)                 ;SET UP THE FORCE ERROR BIT IN
3487                                                                          ;THE MAINTENANCE REGISTER.
3488  016306  005012                          CLR     (R2)                    ;THE FETCH OF THIS INSTRUCTION
3489                                                                          ;SHOULD RESULT IN A PARITY ERROR
3490                                                                          ;ON THE MAIN MEMORY ADDRESS AND CONTROL
3491                                                                          ;LINES. BECAUSE THIS REFERENCE
3492                                                                          ;IS BEING MADE OVER THE UNIBUS
3493                                                                          ;A UNIBUS TIME OUT WILL OCCUR
3494                                                                          ;RESULTING IN AN ABORT TO VECTOR
3495                                                                          ;ERRVEC. THEN IMMEDIATELY FOLLOWING
3496                                                                          ;THIS ABORT TO ERRVEC, THE
3497                                                                          ;PARITY ERROR WILL CAUSE A TRAP
3498                                                                          ;TO CACHVEC!!!
```

```
3499
3500  016310                          MR1:                                    ;REPORT FAILURE OF THE MAINTENANCE
3501  016310  012737  000002  001236        MOV     #2,$TMP2                  ;TO FORCE THE ERROR.
3502  016316  104127                1$:     ERROR   127
3503  016320  012737  177777  031102        MOV     #-1,MANFL2
3504  016326  000503                        BR      MRDONE
3505
3506  016330  022766  177777  000010  MRERRO: CMP    #-1,10(SP)                ;DID 2 TRAPS OCCUR? SEE WHERE
3507                                                                          ;THE MARKER IS ON THE STACK!
3508  016336  001401                        BEQ     MR2
3509  016340  104000                        ERROR
3510
3511  016342  022737  002402  177744  MR2:    CMP    #2402,@#MEMERR            ;DID THE ERROR REGISTER GET
3512  016350  001430                        BEQ     MR3                       ;SET CORRECTLY.
3513
3514                                                                          ;IF NOT REPORT THE ERROR.
3515  016352  022626                        CMP     (SP)+,(SP)+
3516  016354  012637  001236                MOV     (SP)+,$TMP2
3517  016360  022626                        CMP     (SP)+,(SP)+
3518  016362  013737  177740  001240        MOV     @#LOADRS,$TMP3
3519  016370  013737  177742  001242        MOV     @#HIADRS,$TMP4
3520  016376  012737  000002  001244        MOV     #2,$TMP5
3521  016404  012737  002402  001246        MOV     #2402,$TMP6
3522  016412  013737  177744  001250        MOV     @#MEMERR,$TMP7
3523  016420  104131                1$:     ERROR   131
3524  016422  012737  177777  031102        MOV     #-1,MANFL2
3525  016430  000402                        BR      MR4
3526
3527  016432  062706  000012          MR3:  ADD     #12,SP                    ;RESET THE STACK.
3528
3529  016436  005037  177572          MR4:  CLR     @#MMR0
3530  016442  005037  172516                CLR     @#MMR3
3531  016446  012737  177777  177744        MOV     #-1,@#MEMERR              ;TRY TO CLR THE ERROR REG.
3532  016454  005737  177744                TST     @#MEMERR
3533  016460  001416                        BEQ     MR6
3534
3535  016462                          MR5:                                    ;THE ERROR REGISTER WON'T CLR.
3536  016462  013737  177740  001236        MOV     @#LOADRS,$TMP2
3537  016470  013737  177742  001240        MOV     @#HIADRS,$TMP3
3538  016476  013737  177744  001242        MOV     @#MEMERR,$TMP4
3539  016504  104130                1$:     ERROR   130
3540  016506  012737  177777  031062        MOV     #-1,MMRFLG
3541  016514  000410                        BR      MRDONE
3542
3543  016516  022737  177740  177740  MR6:    CMP    #177740,@#LOADRS          ;SEE IF THE ADDRESS REGISTER
3544  016524  001356                        BNE     MR5                       ;GOT RESET.
3545  016526  022737  000003  177742        CMP     #3,@#HIADRS
3546  016534  001352                        BNE     MR5
3547
3548  016536  104410                MRDONE: RSET
3549
3550                                  ;;*****************************************************************
3551                                  ;*TEST 35       CACHE MAINTENANCE AND ERROR REGISTERS TEST 21
3552                                  ;*
3553                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3554                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
```

```
                                        ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
                                        ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
                                        ;*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN EVEN WORD IN THE
                                        ;*PAIR, WHICH IS ALSO THE WANTED WORD.
                                        ;*
                                        ;;**************************************************************
016540  000004                          TST35:  SCOPE
016542  012737  000040  001302                  MOV     #40.,$TIMES     ;;DO 40 ITERATIONS
        000035                          MS=$TN-1
                                                                        ;SET THE SKAD REGISTER
016550  012737  017160  030646                  MOV     #TST36,SKAD     ;IN CASE THE TEST ABORTS.

016556  113737  001102  001232                  MOVB    $TSTNM,$TMPO

016564  104415                                  SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
016566  104416                                  SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
016570  104417                                  SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
016572  104420                                  SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
016574  104412                                  MMSKIP
016576  012737  016756  000114                  MOV     #MSERRO,@#CACHVEC       ;SET UP FOR THE ERROR

016604  012700  172340                          MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                        ;TO RELOCATE EVERYTHING
016610  012702  172300                          MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
016614  012703  000007                          MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
016620  005004                                  CLR     R4              ;BY PASSIVELY IS MEANT
016622  012705  170200                          MOV     #MAPLOO,R5      ;THAT ADDRESS ARE
                                                                        ;RELOCATED TO THEMSELVES.
016626  012722  077406          64$:            MOV     #77406,(R2)+
016632  010401                                  MOV     R4,R1
016634  072127  000006                          ASH     #6,R1
016640  010125                                  MOV     R1,(R5)+
016642  005025                                  CLR     (R5)+
016644  010410                                  MOV     R4,(R0)
016646  062720  170000                          ADD     #170000,(R0)+
016652  062704  000200                          ADD     #200,R4
016656  077315                                  SOB     R3,64$
016660  012710  177600                          MOV     #177600,(R0)
016664  012712  077406                          MOV     #77406,(R2)

016670  012737  000060  172516                  MOV     #60,@#MMR3              ;TURN THE MAP AND ENABLE
016676  012737  000001  177572                  MOV     #1,@#MMR0               ;22 BIT MODE ADDRESSING.
016704  012704  010000                          MOV     #10000,R4               ;PATTERN FOR THE MAINTENANCE
016710  012702  177750                          MOV     #MAINT,R2               ;REGISTER.
016714  012737  000014  177746                  MOV     #M1M0,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
016722  000402                                  BR      MS1

        016724                                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        016724                                  LOC=-4&LOC
        016730                                  LOC=LOC+4
        016730                                  .=LOC

016730  000240                  MS1:            NOP
016732  010412                                  MOV     R4,(R2)                 ;TURN ON THE MAINTENANCE REGISTER.
016734  005701                  MS2:            TST     R1
016736  005012                                  CLR     (R2)
```

```

016740                                  MS3:                                    ;REPORT ERROR.  MAINTENANCE
016740  010437  001236                          MOV     R4,$TMP2                ;FUNCTION FAILED TO
                                                                                ;CAUSE ERROR.
016744  104127                          1$:     ERROR   127
016746  012737  177777  031102          MOV     #-1,MANFL2
016754  000500                                  BR      MSDONE

016756  022737  023404  177744          MSERRO: CMP     #23404,@#MEMERR ;DID THE ERROR REGISTER
016764  001042                                  BNE     69$             ;SET PROPERLY?

016766  022626                          64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
016770  005037  177572                  65$:    CLR     @#MMR0
016774  005037  172516                          CLR     @#MMR3
017000  012737  177777  177744                  MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
017006  005737  177744                          TST     @#MEMERR        ;REGISTER.
017012  001416                                  BEQ     68$

017014                                  66$:                            ;ERROR REGISTER WON'T
017014  013737  177740  001236                  MOV     @#LOADRS,$TMP2  ;CLEAR
017022  013737  177742  001240                  MOV     @#HIADRS,$TMP3
017030  013737  177744  001242                  MOV     @#MEMERR,$TMP4

017036  104130                          67$:    ERROR   130
017040  012737  177777  031062                  MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
017046  000443                                  BR      MSDONE

017050  022737  177740  177740          68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
017056  001356                                  BNE     66$             ;UNLOCKED.
017060  022737  000003  177742                  CMP     #3,@#HIADRS
017066  001352                                  BNE     66$
017070  000432                                  BR      MSDONE

017072                                  69$:                            ;REPORT ERROR REGISTER
017072  012637  001236                          MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
017076  005726                                  TST     (SP)+           ;RESET THE STACK.
017100  013737  177740  001240                  MOV     @#LOADRS,$TMP3
017106  013737  177742  001242                  MOV     @#HIADRS,$TMP4
017114  012737  010000  001244                  MOV     #10000,$TMP5
017122  012737  023404  001246                  MOV     #23404,$TMP6
017130  013737  177744  001250                  MOV     @#MEMERR,$TMP7

017136  104131                          70$:    ERROR   131
017140  012737  177777  031102                  MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
017146  012737  177777  031076                  MOV     #-1,MMRFL2
017154  000705                                  BR      65$
017156  104410                          MSDONE: RSET

                                        ;;*************************************************************
                                        ;*TEST 36       CACHE MAINTENANCE AND ERROR REGISTERS TEST 22
                                        ;*
                                        ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
                                        ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
                                        ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
                                        ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A MAIN MEMORY DATA
                                        ;*PARITY ERROR ON THAT REFERENCE WHICH IS TO AN ODD WORD IN THE
```

```
                                        ;*PAIR, WHICH IS ALSO THE WANTED WORD.
                                        ;*
                                        ;;*****************************************************************
017160  000004                        TST36:  SCOPE
017162  012737  000040  001302                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000036                  MT=$TN-1
                                                                        ;SET THE SKAD REGISTER
017170  012737  017604  030646                MOV     #TST37,SKAD     ;IN CASE THE TEST ABORTS.

017176  113737  001102  001232                MOVB    $TSTNM,$TMP0

017204  104415                               SKPBER                    ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
017206  104416                               SKPBCN                    ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
017210  104417                               SKPBMN                    ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
017212  104420                               SKPBHM                    ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
017214  104412                               MMSKIP

017216  012700  172340                       MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                        ;TO RELOCATE EVERYTHING
017222  012702  172300                       MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
017226  012703  000007                       MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
017232  005004                               CLR     R4              ;BY PASSIVELY IS MEANT
017234  012705  170200                       MOV     #MAPLOD,R5      ;THAT ADDRESS ARE
                                                                        ;RELOCATED TO THEMSELVES.
017240  012722  077406          64$:        MOV     #77406,(R2)+
017244  010401                               MOV     R4,R1
017246  072127  000006                       ASH     #6,R1
017252  010125                               MOV     R1,(R5)+
017254  005025                               CLR     (R5)+
017256  010410                               MOV     R4,(R0)
017260  062720  170000                       ADD     #170000,(R0)+
017264  062704  000200                       ADD     #200,R4
017270  077315                               SOB     R3,64$
017272  012710  177600                       MOV     #177600,(R0)
017276  012712  077406                       MOV     #77406,(R2)

017302  012737  000060  172516               MOV     #60,@#MMR3      ;TURN ON THE MAP AND 22-BIT
017310  012737  000001  177572               MOV     #1,@#MMR0       ;MODE ADDRESSING.
017316  012737  017402  000114               MOV     #MTERR0,@#CACHVEC       ;SET UP FOR THE ERROR.
017324  012737  000014  177746               MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
017332  012704  040000                       MOV     #40000,R4       ;PATTERN TO BE PUT IN MAINT.
017336  012702  177750                       MOV     #MAINT,R2       ;REG.
017342  000403                               BR      MT1

                017344                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                017344                  LOC=-4&LOC
                017350                  LOC=LOC+4
                017350                  .=LOC

017350  000240                          NOP
017352  000240                  MT1:    NOP                     ;NOP FOR SCOPING WITH AN OSCILLOSCOPE!!
017354  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
017356  005701                          TST     R1              ;THE REFERENCE TO THIS INSTRUCTION SHOULD CAUSE A PARITY
017360  005012                          CLR     (R2)            ;ABORT CAUSED BY DETECTION OF BAD PARITY ON
017362  000240                          NOP                     ;THE WANTED, ODD, WORD IN THIS PAIR.
```

```
3723
3724  017364                          MT2:                                ;REPORT ERROR.  MAINTENANCE
3725  017364  010437  001236                MOV     R4,$TMP2              ;FUNCTION FAILED TO
3726                                                                      ;CAUSE ERROR.
3727  017370  104127                  1$:     ERROR   127
3728  017372  012737  177777  031102  MOV     #-1,MANFL2
3729  017400  000500                  BR      MTDONE
3730
3731  017402  022737  023410  177744  MTERR0: CMP     #23410,@#MEMERR       ;DID THE ERROR REGISTER
3732  017410  001042                  BNE     69$                   ;SET PROPERLY?
3733
3734  017412  022626                  64$:    CMP     (SP)+,(SP)+           ;RESET THE STACK
3735  017414  005037  177572          65$:    CLR     @#MMR0
3736  017420  005037  172516                  CLR     @#MMR3
3737  017424  012737  177777  177744          MOV     #-1,@#MEMERR          ;TRY TO CLEAR THE ERROR
3738  017432  005737  177744                  TST     @#MEMERR              ;REGISTER.
3739  017436  001416                  BEQ     68$
3740
3741  017440                          66$:                                ;ERROR REGISTER WON'T
3742  017440  013737  177740  001236          MOV     @#LOADRS,$TMP2        ;CLEAR
3743  017446  013737  177742  001240          MOV     @#HIADRS,$TMP3
3744  017454  013737  177744  001242          MOV     @#MEMERR,$TMP4
3745
3746  017462  104130                  67$:    ERROR   130
3747  017464  012737  177777  031062          MOV     #-1,MMRFLG            ;SIGNAL BAD REGISTER
3748  017472  000443                  BR      MTDONE
3749
3750  017474  022737  177740  177740  68$:    CMP     #177740,@#LOADRS      ;SEE IF ADDRESS REGISTER
3751  017502  001356                  BNE     66$                   ;UNLOCKED.
3752  017504  022737  000003  177742          CMP     #3,@#HIADRS
3753  017512  001352                  BNE     66$
3754  017514  000432                  BR      MTDONE
3755
3756  017516                          69$:                                ;REPORT ERROR REGISTER
3757  017516  012637  001236                  MOV     (SP)+,$TMP2           ;NOT SET AS EXPECTED.
3758  017522  005726                  TST     (SP)+                 ;RESET THE STACK.
3759  017524  013737  177740  001240          MOV     @#LOADRS,$TMP3
3760  017532  013737  177742  001242          MOV     @#HIADRS,$TMP4
3761  017540  012737  040000  001244          MOV     #40000,$TMP5
3762  017546  012737  023410  001246          MOV     #23410,$TMP6
3763  017554  013737  177744  001250          MOV     @#MEMERR,$TMP7
3764
3765  017562  104131                  70$:    ERROR   131
3766  017564  012737  177777  031102          MOV     #-1,MANFL2            ;SIGNAL BAD REGISTER
3767  017572  012737  177777  031076          MOV     #-1,MMRFL2
3768  017600  000705                  BR      65$
3769  017602  104410                  MTDONE: RSET
3770
3771                                  ;;*****************************************************************
3772                                  ;*TEST 37       CACHE MAINTENANCE AND ERROR REGISTERS TEST 23
3773                                  ;*
3774                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3775                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3776                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3777                                  ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY
3778                                  ;*PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE
```

```
                                        ;*LOW BYTE OF THAT ADDRESS .
                                        ;*
                                        ;;*****************************************************************
017604  000004                  TST37:  SCOPE
017606  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000037          MU=$TN-1
                                                                ;SET THE SKAD REGISTER
017614  012737  020224  030646          MOV     #TST40,SKAD     ;IN CASE THE TEST ABORTS.

017622  113737  001102  001232          MOVB    $TSTNM,$TMP0

017630  104415                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
017632  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
017634  104417                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
017636  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
017640  104412                          MMSKIP

017642  012700  172340                  MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                ;TO RELOCATE EVERYTHING
017646  012702  172300                  MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
017652  012703  000007                  MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
017656  005004                          CLR     R4              ;BY PASSIVELY IS MEANT
017660  012705  170200                  MOV     #MAPL00,R5      ;THAT ADDRESS ARE
                                                                ;RELOCATED TO THEMSELVES.
017664  012722  077406          64$:    MOV     #77406,(R2)+
017670  010401                          MOV     R4,R1
017672  072127  000006                  ASH     #6,R1
017676  010125                          MOV     R1,(R5)+
017700  005025                          CLR     (R5)+
017702  010410                          MOV     R4,(R0)
017704  062720  170000                  ADD     #170000,(R0)+
017710  062704  000200                  ADD     #200,R4
017714  077315                          SOB     R3,64$
017716  012710  177600                  MOV     #177600,(R0)
017722  012712  077406                  MOV     #77406,(R2)

017726  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
017734  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
017742  012737  020022  000114          MOV     #MUERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
017750  012737  000030  177746          MOV     #SOM1,@#CONTRL  ;SELECT GROUP ADDRESS
017756  012704  000400                  MOV     #400,R4 ;PATTERN TO BE LOADED IN THE
017762  012702  177750                  MOV     #MAINT,R2       ;MAINTENANCE REG.
017766  000403                          BR      MU1

        017770                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        017770                  LOC=-4&LOC
        017774                  LOC=LOC+4
        017774                  .=LOC

017774  000240                  MU1:    NOP
017776  000240                          NOP
020000  010412                          MOV     R4,(R2)         ;SET THE MAINT REG.
020002  005012                          CLR     (R2)            ;THIS FETCH SHOULD CAUSE
                                                                ;A PARITY ERROR IN GROUP
                                                                ;ADDRESS 0 MEMORY
```

```
3835  020004                          MU2:                                    ;REPORT ERROR.  MAINTENANCE
3836  020004  010437  001236                  MOV     R4,$TMP2                ;FUNCTION FAILED TO
3837                                                                          ;CAUSE ERROR.
3838  020010  104127                  1$:     ERROR   127
3839  020012  012737  177777  031102  MOV     #-1,MANFL2
3840  020020  000500                          BR      MUDONE
3841
3842  020022  022737  002420  177744  MUERRO: CMP     #2420,@#MEMERR          ;DID THE ERROR REGISTER
3843  020030  001042                          BNE     69$                     ;SET PROPERLY?
3844
3845  020032  022626                  64$:    CMP     (SP)+,(SP)+             ;RESET THE STACK
3846  020034  005037  177572          65$:    CLR     @#MMR0
3847  020040  005037  172516                  CLR     @#MMR3
3848  020044  012737  177777  177744          MOV     #-1,@#MEMERR            ;TRY TO CLEAR THE ERROR
3849  020052  005737  177744                  TST     @#MEMERR                ;REGISTER.
3850  020056  001416                          BEQ     68$
3851
3852  020060                          66$:                                    ;ERROR REGISTER WON'T
3853  020060  013737  177740  001236          MOV     @#LOADRS,$TMP2          ;CLEAR
3854  020066  013737  177742  001240          MOV     @#HIADRS,$TMP3
3855  020074  013737  177744  001242          MOV     @#MEMERR,$TMP4
3856
3857  020102  104130                  67$:    ERROR   130
3858  020104  012737  177777  031062          MOV     #-1,MMRFLG              ;SIGNAL BAD REGISTER
3859  020112  000443                          BR      MUDONE
3860
3861  020114  022737  177740  177740  68$:    CMP     #177740,@#LOADRS        ;SEE IF ADDRESS REGISTER
3862  020122  001356                          BNE     66$                     ;UNLOCKED.
3863  020124  022737  000003  177742          CMP     #3,@#HIADRS
3864  020132  001352                          BNE     66$
3865  020134  000432                          BR      MUDONE
3866
3867  020136                          69$:                                    ;REPORT ERROR REGISTER
3868  020136  012637  001236                  MOV     (SP)+,$TMP2             ;NOT SET AS EXPECTED.
3869  020142  005726                          TST     (SP)+                   ;RESET THE STACK.
3870  020144  013737  177740  001240          MOV     @#LOADRS,$TMP3
3871  020152  013737  177742  001242          MOV     @#HIADRS,$TMP4
3872  020160  012737  000400  001244          MOV     #400,$TMP5
3873  020166  012737  002420  001246          MOV     #2420,$TMP6
3874  020174  013737  177744  001250          MOV     @#MEMERR,$TMP7
3875
3876  020202  104131                  70$:    ERROR   131
3877  020204  012737  177777  031102          MOV     #-1,MANFL2              ;SIGNAL BAD REGISTER
3878  020212  012737  177777  031076          MOV     #-1,MMRFL2
3879  020220  000705                          BR      65$
3880  020222  104410                  MUDONE: RSET
3881
3882                                  ;;*****************************************************************
3883                                  ;*TEST 40        CACHE MAINTENANCE AND ERROR REGISTERS TEST 24
3884                                  ;*
3885                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3886                                  ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3887                                  ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3888                                  ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE ADDRESS MEMORY
3889                                  ;*PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE
3890                                  ;*LOW BYTE OF THAT ADDRESS .
```

```
3891                                    ;*
3892                                    ;;**********************************************************************
3893  020224  000004                    TST40:  SCOPE
3894  020226  012737  000040  001302            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
3895          000040                    MV=$TN-1
3896                                                                    ;SET THE SKAD REGISTER
3897  020234  012737  020644  030646            MOV     #TST41,SKAD     ;IN CASE THE TEST ABORTS.
3898
3899  020242  113737  001102  001232            MOVB    $TSTNM,$TMP0
3900
3901  020250  104415                            SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
3902  020252  104416                            SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
3903  020254  104417                            SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
3904  020256  104420                            SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
3905  020260  104412                            MMSKIP
3906
3907  020262  012700  172340                    MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
3908                                                                    ;TO RELOCATE EVERYTHING
3909  020266  012702  172300                    MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
3910  020272  012703  000007                    MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
3911  020276  005004                            CLR     R4              ;BY PASSIVELY IS MEANT
3912  020300  012705  170200                    MOV     #MAPL00,R5      ;THAT ADDRESS ARE
3913                                                                    ;RELOCATED TO THEMSELVES.
3914  020304  012722  077406            64$:    MOV     #77406,(R2)+
3915  020310  010401                            MOV     R4,R1
3916  020312  072127  000006                    ASH     #6,R1
3917  020316  010125                            MOV     R1,(R5)+
3918  020320  005025                            CLR     (R5)+
3919  020322  010410                            MOV     R4,(R0)
3920  020324  062720  170000                    ADD     #170000,(R0)+
3921  020330  062704  000200                    ADD     #200,R4
3922  020334  077315                            SOB     R3,64$
3923  020336  012710  177600                    MOV     #177600,(R0)
3924  020342  012712  077406                    MOV     #77406,(R2)
3925
3926  020346  012737  000060  172516            MOV     #60,@#MMR3      ;TURN ON THE MAP AND
3927  020354  012737  000001  177572            MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
3928  020362  012737  020442  000114            MOV     #MVERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
3929  020370  012737  000044  177746            MOV     #S1M0,@#CONTRL  ;SELECT GROUP ADDRESS
3930  020376  012704  002000                    MOV     #2000,R4        ;PATTERN TO BE LOADED IN THE
3931  020402  012702  177750                    MOV     #MAINT,R2       ;MAINTENANCE REG.
3932  020406  000403                            BR      MV1
3933
3934          020410                    LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
3935          020410                    LOC=-4&LOC
3936          020414                    LOC=LOC+4
3937          020414                    .=LOC
3938
3939  020414  000240                            NOP
3940  020416  000240                    MV1:    NOP
3941  020420  010412                            MOV     R4,(R2)         ;SET THE MAINT REG.
3942  020422  005012                            CLR     (R2)            ;THIS FETCH SHOULD CAUSE
3943                                                                    ;A PARITY ERROR IN GROUP
3944                                                                    ;ADDRESS 1 MEMORY
3945
3946  020424                            MV2:                            ;REPORT ERROR.  MAINTENANCE
```

```
3947  020424  010437  001236                MOV     R4,$TMP2        ;FUNCTION FAILED TO
3948                                                                  ;CAUSE ERROR.
3949  020430  104127                1$:     ERROR   127
3950  020432  012737  177777  031102  MOV   #-1,MANFL2
3951  020440  000500                        BR      MVDONE
3952
3953  020442  022737  002440  177744  MVERRO: CMP     #2440,@#MEMERR  ;DID THE ERROR REGISTER
3954  020450  001042                        BNE     69$             ;SET PROPERLY?
3955
3956  020452  022626                64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
3957  020454  005037  177572        65$:    CLR     @#MMR0
3958  020460  005037  172516                CLR     @#MMR3
3959  020464  012737  177777  177744        MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
3960  020472  005737  177744                TST     @#MEMERR        ;REGISTER.
3961  020476  001416                        BEQ     68$
3962
3963  020500                        66$:                            ;ERROR REGISTER WON'T
3964  020500  013737  177740  001236        MOV     @#LOADRS,$TMP2  ;CLEAR
3965  020506  013737  177742  001240        MOV     @#HIADRS,$TMP3
3966  020514  013737  177744  001242        MOV     @#MEMERR,$TMP4
3967
3968  020522  104130                67$:    ERROR   130
3969  020524  012737  177777  031062        MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
3970  020532  000443                        BR      MVDONE
3971
3972  020534  022737  177740  177740  68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
3973  020542  001356                        BNE     66$             ;UNLOCKED.
3974  020544  022737  000003  177742        CMP     #3,@#HIADRS
3975  020552  001352                        BNE     66$
3976  020554  000432                        BR      MVDONE
3977
3978  020556                        69$:                            ;REPORT ERROR REGISTER
3979  020556  012637  001236                MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
3980  020562  005726                        TST     (SP)+           ;RESET THE STACK.
3981  020564  013737  177740  001240        MOV     @#LOADRS,$TMP3
3982  020572  013737  177742  001242        MOV     @#HIADRS,$TMP4
3983  020600  012737  002000  001244        MOV     #2000,$TMP5
3984  020606  012737  002440  001246        MOV     #2440,$TMP6
3985  020614  013737  177744  001250        MOV     @#MEMERR,$TMP7
3986
3987  020622  104131                70$:    ERROR   131
3988  020624  012737  177777  031102        MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
3989  020632  012737  177777  031076        MOV     #-1,MMRFL2
3990  020640  000705                        BR      65$
3991  020642  104410                MVDONE: RSET
3992
3993                                ;;*****************************************************************
3994                                ;*TEST 41        CACHE MAINTENANCE AND ERROR REGISTERS TEST 25
3995                                ;*
3996                                ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
3997                                ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
3998                                ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
3999                                ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY
4000                                ;*PARITY ERROR IN GROUP 0 ON THAT REFERENCE. THE ERROR IS ON THE
4001                                ;*LOW BYTE OF THAT DATA .
4002                                ;*
```

```
4003                                    ;;*****************************************************************
4004  020644  000004                    $TST41:  SCOPE
4005  020646  012737  000040  001302             MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4006          000041                    MW=$TN-1
4007                                                                      ;SET THE SKAD REGISTER
4008  020654  012737  021264  030646             MOV     #TST42,SKAD     ;IN CASE THE TEST ABORTS.
4009
4010  020662  113737  001102  001232             MOVB    $TSTNM,$TMP0
4011
4012  020670  104415                             SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4013  020672  104416                             SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4014  020674  104417                             SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4015  020676  104420                             SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4016  020700  104412                             MMSKIP
4017
4018  020702  012700  172340                     MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
4019                                                                      ;TO RELOCATE EVERYTHING
4020  020706  012702  172300                     MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
4021  020712  012703  000007                     MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
4022  020716  005004                             CLR     R4              ;BY PASSIVELY IS MEANT
4023  020720  012705  170200                     MOV     #MAPLOD,R5      ;THAT ADDRESS ARE
4024                                                                      ;RELOCATED TO THEMSELVES.
4025  020724  012722  077406            64$:    MOV     #77406,(R2)+
4026  020730  010401                             MOV     R4,R1
4027  020732  072127  000006                     ASH     #6,R1
4028  020736  010125                             MOV     R1,(R5)+
4029  020740  005025                             CLR     (R5)+
4030  020742  010410                             MOV     R4,(R0)
4031  020744  062720  170000                     ADD     #170000,(R0)+
4032  020750  062704  000200                     ADD     #200,R4
4033  020754  077315                             SOB     R3,64$
4034  020756  012710  177600                     MOV     #177600,(R0)
4035  020762  012712  077406                     MOV     #77406,(R2)
4036
4037  020766  012737  000060  172516             MOV     #60,@#MMR3      ;TURN ON THE MAP AND
4038  020774  012737  000001  177572             MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
4039  021002  012737  021062  000114             MOV     #MWERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
4040  021010  012737  000030  177746             MOV     #SOM1,@#CONTRL  ;SELECT GROUP DATA
4041  021016  012704  000020                     MOV     #20,R4  ;PATTERN TO BE LOADED IN THE
4042  021022  012702  177750                     MOV     #MAINT,R2       ;MAINTENANCE REG.
4043  021026  000403                             BR      MW1
4044
4045          021030                             LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4046          021030                             LOC=-4&LOC
4047          021034                             LOC=LOC+4
4048          021034                             .=LOC
4049
4050  021034  000240                             NOP
4051  021036  000240                    MW1:     NOP
4052  021040  010412                             MOV     R4,(R2)         ;SET THE MAINT REG.
4053  021042  005012                             CLR     (R2)            ;THIS FETCH SHOULD CAUSE
4054                                                                      ;A PARITY ERROR IN GROUP
4055                                                                      ;DATA 0 MEMORY
4056
4057  021044                            MW2:                              ;REPORT ERROR.  MAINTENANCE
4058  021044  010437  001236                     MOV     R4,$TMP2        ;FUNCTION FAILED TO
```

```
4059                                                                    ;CAUSE ERROR.
4060  021050  104127                    1$:     ERROR   127
4061  021052  012737  177777  031102    MOV     #-1,MANFL2
4062  021060  000500                    BR      MWDONE
4063
4064  021062  022737  002500  177744  MWERRO: CMP     #2500,@#MEMERR  ;DID THE ERROR REGISTER
4065  021070  001042                    BNE     69$             ;SET PROPERLY?
4066
4067  021072  022626                    64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
4068  021074  005037  177572            65$:    CLR     @#MMR0
4069  021100  005037  172516                    CLR     @#MMR3
4070  021104  012737  177777  177744            MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
4071  021112  005737  177744                    TST     @#MEMERR        ;REGISTER.
4072  021116  001416                    BEQ     68$
4073
4074  021120                            66$:                            ;ERROR REGISTER WON'T
4075  021120  013737  177740  001236            MOV     @#LOADRS,$TMP2  ;CLEAR
4076  021126  013737  177742  001240            MOV     @#HIADRS,$TMP3
4077  021134  013737  177744  001242            MOV     @#MEMERR,$TMP4
4078
4079  021142  104130                    67$:    ERROR   130
4080  021144  012737  177777  031062            MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
4081  021152  000443                    BR      MWDONE
4082
4083  021154  022737  177740  177740  68$:     CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
4084  021162  001356                    BNE     66$                     ;UNLOCKED.
4085  021164  022737  000003  177742            CMP     #3,@#HIADRS
4086  021172  001352                    BNE     66$
4087  021174  000432                    BR      MWDONE
4088
4089  021176                            69$:                            ;REPORT ERROR REGISTER
4090  021176  012637  001236                    MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
4091  021202  005726                    TST     (SP)+                   ;RESET THE STACK.
4092  021204  013737  177740  001240            MOV     @#LOADRS,$TMP3
4093  021212  013737  177742  001242            MOV     @#HIADRS,$TMP4
4094  021220  012737  000020  001244            MOV     #20,$TMP5
4095  021226  012737  002500  001246            MOV     #2500,$TMP6
4096  021234  013737  177744  001250            MOV     @#MEMERR,$TMP7
4097
4098  021242  104131                    70$:    ERROR   131
4099  021244  012737  177777  031102            MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
4100  021252  012737  177777  031076            MOV     #-1,MMRFL2
4101  021260  000705                    BR      65$
4102  021262  104410                    MWDONE: RSET
4103
4104                                    ;;*****************************************************************
4105                                    ;*TEST 42        CACHE MAINTENANCE AND ERROR REGISTERS TEST 26
4106                                    ;*
4107                                    ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO SET CORRECTLY
4108                                    ;*AS THE RESULT OF A CPU REFERENCE WHICH RELOCATED THROUGH THE MEMORY
4109                                    ;*MANAGEMENT UNIT TO THE UNIBUS AND THROUGH THE UNIBUS MAP TO THE CACHE.
4110                                    ;*THE MAINTENANCE REGISTER IS USED TO CAUSE A CACHE DATA MEMORY
4111                                    ;*PARITY ERROR IN GROUP 1 ON THAT REFERENCE. THE ERROR IS ON THE
4112                                    ;*LOW BYTE OF THAT DATA .
4113                                    ;*
4114                                    ;;*****************************************************************
```

```
4115  021264  000004                      TST42:  SCOPE
4116  021266  012737  000040  001302              MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4117          000042                      MX=$TN-1
4118                                                                      ;SET THE SKAD REGISTER
4119  021274  012737  021704  030646              MOV     #TST43,SKAD     ;IN CASE THE TEST ABORTS.
4120
4121  021302  113737  001102  001332              MOVB    $TSTNM,$TMPO
4122
4123  021310  104415                              SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4124  021312  104416                              SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4125  021314  104417                              SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4126  021316  104420                              SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4127  021320  104412                              MMSKIP
4128
4129  021322  012700  172340                      MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
4130                                                                      ;TO RELOCATE EVERYTHING
4131  021326  012702  172300                      MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
4132  021332  012703  000007                      MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
4133  021336  005004                              CLR     R4              ;BY PASSIVELY IS MEANT
4134  021340  012705  170200                      MOV     #MAPLOO,R5      ;THAT ADDRESS ARE
4135                                                                      ;RELOCATED TO THEMSELVES.
4136  021344  012722  077406              64$:    MOV     #77406,(R2)+
4137  021350  010401                              MOV     R4,R1
4138  021352  072127  000006                      ASH     #6,R1
4139  021356  010125                              MOV     R1,(R5)+
4140  021360  005025                              CLR     (R5)+
4141  021362  010410                              MOV     R4,(R0)
4142  021364  062720  170000                      ADD     #170000,(R0)+
4143  021370  062704  000200                      ADD     #200,R4
4144  021374  077315                              SOB     R3,64$
4145  021376  012710  177600                      MOV     #177600,(R0)
4146  021402  012712  077406                      MOV     #77406,(R2)
4147
4148  021406  012737  000060  172516              MOV     #60,@#MMR3      ;TURN ON THE MAP AND
4149  021414  012737  000001  177572              MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
4150  021422  012737  021502  000114              MOV     #MXERR0,@#CACHVEC       ;SETUP FOR THE ERROR.
4151  021430  012737  000044  177746              MOV     #S1M0,@#CONTRL  ;SELECT GROUP DATA
4152  021436  012704  000100                      MOV     #100,R4 ;PATTERN TO BE LOADED IN THE
4153  021442  012702  177750                      MOV     #MAINT,R2       ;MAINTENANCE REG.
4154  021446  000403                              BR      MX1
4155
4156          021450                              LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4157          021450                              LOC=-4&LOC
4158          021454                              LOC=LOC+4
4159          021454                              .=LOC
4160
4161  021454  000240                              NOP
4162  021456  000240                      MX1:    NOP
4163  021460  010412                              MOV     R4,(R2)         ;SET THE MAINT REG.
4164  021462  005012                              CLR     (R2)            ;THIS FETCH SHOULD CAUSE
4165                                                                      ;A PARITY ERROR IN GROUP
4166                                                                      ;DATA 1 MEMORY
4167
4168  021464                              MX2:                            ;REPORT ERROR.  MAINTENANCE
4169  021464  010437  001236                      MOV     R4,$TMP2        ;FUNCTION FAILED TO
4170                                                                      ;CAUSE ERROR.
```

```
 4171  021470  104127                    1$:     ERROR   127
 4172  021472  012737  177777  031102    MOV     #-1,MANFL2
 4173  021500  000500                    BR      MXDONE
 4174
 4175  021502  022737  002600  177744    MXERR0: CMP     #2600,@#MEMERR  ;DID THE ERROR REGISTER
 4176  021510  001042                    BNE     69$             ;SET PROPERLY?
 4177
 4178  021512  022626                    64$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
 4179  021514  005037  177572            65$:    CLR     @#MMR0
 4180  021520  005037  172516            CLR     @#MMR3
 4181  021524  012737  177777  177744    MOV     #-1,@#MEMERR    ;TRY TO CLEAR THE ERROR
 4182  021532  005737  177744            TST     @#MEMERR        ;REGISTER.
 4183  021536  001416                    BEQ     68$
 4184
 4185  021540                            66$:                    ;ERROR REGISTER WON'T
 4186  021540  013737  177740  001236    MOV     @#LOADRS,$TMP2  ;CLEAR
 4187  021546  013737  177742  001240    MOV     @#HIADRS,$TMP3
 4188  021554  013737  177744  001242    MOV     @#MEMERR,$TMP4
 4189
 4190  021562  104130                    67$:    ERROR   130
 4191  021564  012737  177777  031062    MOV     #-1,MMRFLG      ;SIGNAL BAD REGISTER
 4192  021572  000443                    BR      MXDONE
 4193
 4194  021574  022737  177740  177740    68$:    CMP     #177740,@#LOADRS ;SEE IF ADDRESS REGISTER
 4195  021602  001356                    BNE     66$             ;UNLOCKED.
 4196  021604  022737  000003  177742    CMP     #3,@#HIADRS
 4197  021612  001352                    BNE     66$
 4198  021614  000432                    BR      MXDONE
 4199
 4200  021616                            69$:                    ;REPORT ERROR REGISTER
 4201  021616  012637  001236            MOV     (SP)+,$TMP2     ;NOT SET AS EXPECTED.
 4202  021622  005726                    TST     (SP)+           ;RESET THE STACK.
 4203  021624  013737  177740  001240    MOV     @#LOADRS,$TMP3
 4204  021632  013737  177742  001242    MOV     @#HIADRS,$TMP4
 4205  021640  012737  000100  001244    MOV     #100,$TMP5
 4206  021646  012737  002600  001246    MOV     #2600,$TMP6
 4207  021654  013737  177744  001250    MOV     @#MEMERR,$TMP7
 4208
 4209  021662  104131                    70$:    ERROR   131
 4210  021664  012737  177777  031102    MOV     #-1,MANFL2      ;SIGNAL BAD REGISTER
 4211  021672  012737  177777  031076    MOV     #-1,MMRFL2
 4212  021700  000705                    BR      65$
 4213  021702  104410                    MXDONE: RSET
 4214
 4215                                    ;;*****************************************************************
 4216                                    ;*TEST 43        CACHE ERROR REGISTER UNIBUS TIME OUT TEST
 4217                                    ;*
 4218                                    ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO COMPREHEND A
 4219                                    ;*CPU TO UNIBUS THROUGH THE MAP TO THE CACHE REFERENCE WHICH
 4220                                    ;*TIMES OUT IN MAIN MEMORY. MANY SUCH NON-EXISTENT MEMORY LOCATIONS
 4221                                    ;*ARE CONVIENTLY GUARENTEED TO EXIST! ALL THE ADDRESSES
 4222                                    ;*FROM 17000000 THROUGH 17777776 ARE ADDRESSES
 4223                                    ;*WHICH CAN NOT EXIST. HERE ONLY ONE OF THESE ADDRESSES, 17777776,
 4224                                    ;*WILL BE USED TO CAUSE A TIME OUT ON THE UNIBUS AN THE CONSEQUENT
 4225                                    ;*ABORT TO VECTOR ERRVEC.
 4226                                    ;*
```

```
                                        ;;*****************************************************************
000004                          $TST43:  SCOPE
021704  000004
021706  012737  000040  001302          MOV     #40.,$TIMES             ;;DO 40 ITERATIONS
000043                          MQ=:$TN-1
                                                                        ;SET THE SKAD REGISTER
021714  012737  022334  030646          MOV     #TST44,SKAD             ;IN CASE THE TEST ABORTS.

021722  113737  001102  001232          MOVB    $TSTNM,$TMPO
021730  012737  030522  000114          MOV     #SPUR,@#CACHVEC                 ;EXPECT NO PARITY ERRORS.

021736  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
021740  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
021742  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
021744  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
021746  104412                          MMSKIP

021750  012700  172340                  MOV     #KIPAR0,R0      ;INITIALLY PUT MEMORY
021754  012701  077406                  MOV     #77406,R1       ;MANAGEMENT IN A 'PASSIVE'
021760  012702  172300                  MOV     #KIPDR0,R2      ;STATE, THAT IS MAP ALL
021764  012703  000010                  MOV     #10,R3          ;VIRTUAL ADDRESSES ON TO
021770  010122                  64$:    MOV     R1,(R2)+        ;THEMSELVES AS PHYSICAL
021772  077302                          SOB     R3,64$          ;ADDRESSES.
021774  005020                          CLR     (R0)+
021776  012720  000200                  MOV     #200,(R0)+
022002  012720  000400                  MOV     #400,(R0)+
022006  012720  000600                  MOV     #600,(R0)+
022012  012720  001000                  MOV     #1000,(R0)+
022016  012720  001200                  MOV     #1200,(R0)+
022022  012720  001400                  MOV     #1400,(R0)+
022026  012710  177600                  MOV     #177600,(R0)

022032  012737  000060  172516          MOV     #60,@#MMR3              ;TURN ON THE MAPPING BOX
022040  012737  000001  177572          MOV     #1,@#MMR0               ;AND 22 BIT MODE ADDRESSING.
022046  012737  170000  172354          MOV     #170000,@#KIPAR6        ;MAKE KIPAR6 RELOCATE
                                                                        ;TO THE UNIBUS.
022054  012737  022126  000004          MOV     #MQERR,@#ERRVEC         ;SET UP THE TIME OUT VECTOR.

022062  012737  177776  170200          MOV     #-2,@#MAPL00            ;SET THE MAP REGISTER 0
022070  012737  000077  170202          MOV     #77,@#MAPH00
022076  012700  140000                  MOV     #140000,R0              ;THIS IS THE VIRTUAL ADDRESS OF THE
                                                                        ;TEST ADDRESS. IT WILL RELOCATE
                                                                        ;THROUGH KIPAR6 TO THE UNIBUS AS
                                                                        ;A 000000. FROM THE UNIBUS
                                                                        ;IT WILL BE RELOCATED THROUGH
                                                                        ;MAP REGISTER 0 TO THE CACHE WHERE
                                                                        ;IT WILL TRY TO REFERENCE
                                                                        ;17777776, AND HOPEFULLY TIME OUT.
022102  000240                          NOP                             ;FOR SCOPING WITH AN OSCILLOSCOPE!
022104  005710                          TST     (R0)                    ;MAKE THE REFERENCE!

022106                          MQ1:                                    ;NO TIME OUT OCCURRED, REPORT
022106  012737  177776  001236          MOV     #-2,$TMP2               ;THE ERROR.
022114  012737  000077  001240          MOV     #77,$TMP3
022122  104132                  1$:     ERROR   132
022124  000502                          BR      MQDONE
```

```
4283  022126  032737  000020  177766  MQERR:  BIT     #20,@#CPUERR            ;SEE IF A TIME OUT HAS CAUSED
4284  022134  001002                          BNE     MQ2                     ;AN ABORT TO THIS ROUTINE.
4285  022136  000137  030474                  JMP     CPSPUR                  ;IF NOT GO TO THE SPURIOUS
4286                                                                          ;UNEXPECTED, CPU ERROR HANDLER.
4287  022142  022737  000000  177744  MQ2:    CMP     #0,@#MEMERR             ;OTHERWISE SEE IF THE ERROR
4288  022150  001427                          BEQ     MQ3                     ;REGISTER GOT SET CORRECTLY.
4289
4290                                                                  ;IF IT IS NOT SET CORRECTLY REPORT ERROR.
4291  022152  012637  001236                  MOV     (SP)+,$TMP2
4292  022156  005726                          TST     (SP)+
4293  022160  013737  177740  001240          MOV     @#LOADRS,$TMP3
4294  022166  013737  177742  001242          MOV     @#HIADRS,$TMP4
4295  022174  012737  177776  001244          MOV     #-2,$TMP5
4296  022202  012737  000077  001246          MOV     #77,$TMP6
4297  022210  013737  177744  001250          MOV     @#MEMERR,$TMP7
4298  022216  104133                  1$:     ERROR   133
4299  022220  012737  177777  031076          MOV     #-1,MMRFL2
4300  022226  000401                          BR      MQ4
4301
4302  022230  022626                  MQ3:    CMP     (SP)+,(SP)+             ;RESET THE STACK
4303
4304  022232  005037  177572          MQ4:    CLR     @#MMR0
4305  022236  005037  172516                  CLR     @#MMR3
4306  022242  012737  177777  177744          MOV     #-1,@#MEMERR            ;TRY TO CLEAR THE ERROR REGISTER.
4307  022250  005737  177744                  TST     @#MEMERR
4308  022254  001416                          BEQ     MQ6
4309
4310  022256                  MQ5:                                    ;REPORT THE FAILURE OF THE ERROR
4311  022256  013737  177740  001236          MOV     @#LOADRS,$TMP2          ;REGISTER TO CLEAR!
4312  022264  013737  177742  001240          MOV     @#HIADRS,$TMP3
4313  022272  013737  177744  001242          MOV     @#MEMERR,$TMP4
4314  022300  104130                  1$:     ERROR   130
4315  022302  012737  177777  031062          MOV     #-1,MMRFLG
4316  022310  000410                          BR      MQDONE
4317
4318  022312  022737  177740  177740  MQ6:    CMP     #177740,@#LOADRS        ;SEE IF THE ADDRESS REGISTER
4319  022320  001356                          BNE     MQ5                     ;GOT RESET.
4320  022322  022737  000003  177742          CMP     #3,@#HIADRS
4321  022330  001352                          BNE     MQ5
4322
4323  022332  104410          MQDONE: RSET
4324
4325                          ;;**************************************************************
4326                          ;*TEST 44        CACHE CONTROL REGISTER DISABLE TRAPS TEST 1
4327                          ;*
4328                          ;*THIS IS A TEST OF THE CONTROL REGISTER'S ABILITY TO DISABLE A TRAP
4329                          ;*OCCURRING AS THE RESULT OF A MAIN MEMORY DATA PARITY ERROR IN THE
4330                          ;*UNWANTED WORD OF THE REFERENCED PAIR. THE MAINTENANCE REGISTER IS
4331                          ;*USED TO FORCE AN ERROR ON THE LOW BYTE OF THE ODD WORD WHEN REFERENCING
4332                          ;*THE EVEN WORD OF THAT PAIR.
4333                          ;*
4334                          ;;**************************************************************
4335  022334  000004          TST44:  SCOPE
4336  022336  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
4337           000044         KV=$TN-1
4338                                                                  ;SET THE SKAD REGISTER
```

```
 4339  022344  012737  022510  030646          MOV     #TST45,$KAD     ;IN CASE THE TEST ABORTS.
 4340
 4341  022352  113737  001102  001232          MOVB    $TSTNM,$TMP0
 4342
 4343  022360  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
 4344  022362  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
 4345  022364  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
 4346  022366  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
 4347  022370  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
 4348  022376  052737  000001  177746          BIS     #BIT0,@#CONTRL          ;DISABLE 'WARNING' TRAPS.
 4349  022404  012737  022446  000114          MOV     #KVERR,@#CACHVEC                ;SET UP FOR THE ERROR ABOUT TO BE FORCED
 4350  022412  012704  040000                  MOV     #40000,R4               ;PATTERN FOR THE MAINTENANCE
 4351  022416  012702  177750                  MOV     #MAINT,R2               ;REGISTER.
 4352  022422  000402                          BR      KV1
 4353
 4354          022424                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 4355          022424                          LOC=-4&LOC
 4356          022430                          LOC=LOC+4
 4357          022430                          .=LOC
 4358
 4359  022430  000240                  KV1:    NOP
 4360  022432  010412                          MOV     R4,(R2)                 ;SET THE MAINTENANCE REGISTER
 4361  022434  000240                          NOP                             ;WHEN THIS NOP IS FETCHED AN ERROR
 4362  022436  005701                  KV2:    TST     R1                      ;WILL BE RECOGNIZED BECAUSE OF THE
 4363                                                                          ;CONTENTS OF THE LOCATION KV2!
 4364                                                                          ;THIS PARITY ERROR WOULD
 4365                                                                          ;NORMALLY RELUT IN A TRAP BUT
 4366                                                                          ;BECAUSE TRAPS HAVE BEEN DISABLED
 4367                                                                          ;NONE SHOULD OCCUR!!!
 4368  022440  005012                          CLR     (R2)
 4369  022442  000240                          NOP
 4370  022444  000420                          BR      KVDONE                  ;GOOD, NO TRAP OCCURRED!
 4371
 4372  022446                          KVERR:                          ;COME HERE IF A TRAP OCCURS
 4373  022446  012637  001236                  MOV     (SP)+,$TMP2             ;AND REPORT THE ERROR.
 4374  022452  005726                          TST     (SP)+
 4375  022454  013737  177746  001240          MOV     @#CONTRL,$TMP3
 4376  022462  013737  177740  001242          MOV     @#LOADRS,$TMP4
 4377  022470  013737  177742  001244          MOV     @#HIADRS,$TMP5
 4378  022476  013737  177744  001246          MOV     @#MEMERR,$TMP6
 4379  022504  104134                  1$:     ERROR   134
 4380
 4381  022506  104410                  KVDONE: RSET
 4382
 4383
 4384
 4385                                          ;;*****************************************************************
 4386                                          ;*TEST 45       CACHE CONTROL REGISTER DISABLE TRAPS TEST 2
 4387                                          ;*
 4388                                          ;*THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION.
 4389                                          ;*IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE ADDRESS
 4390                                          ;*MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO
 4391                                          ;*FORCE THE ERROR ON THE LOW BYTE OF THE ADDRESS, IN THE ADDRESS MEMORY
 4392                                          ;*OF GROUP 0.
 4393                                          ;*
 4394                                          ;;*****************************************************************
```

```
4395  022510  000004                    TST45:  SCOPE
4396  022512  012737  000040  001302            MOV     #40.,$TIMES         ;;DO 40 ITERATIONS
4397          000045                    KX=$TN-1
4398                                                                        ;SET THE SKAD REGISTER
4399  022520  012737  022710  030646            MOV     #TST46,SKAD         ;IN CASE THE TEST ABORTS.
4400
4401  022526  113737  001102  001232            MOVB    $TSTNM,$TMP0
4402
4403  022534  104415                            SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
4404  022536  104416                            SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
4405  022540  104417                            SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
4406  022542  104420                            SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
4407  022544  012737  000030  177746            MOV     #SDM1,@#CONTRL      ;USE GROUP ZERO
4408  022552  012700  022640                    MOV     #KX2,R0             ;MAKE KX2 A HIT IN GROUP
4409  022556  005710                            TST     (R0)                ;ZERO.
4410  022560  005710                            TST     (R0)
4411
4412                                                                        ;SEE IF REFERENCE ADDRESS
4413  022562  032737  000010  177752            BIT     #10,@#HITMIS        ;IS A HIT.
4414  022570  001007                            BNE     KX1
4415                                                                        ;IF NOT ERROR!
4416  022572  010037  001236                    MOV     R0,$TMP2
4417  022576  012737  000000  001234            MOV     #0,$TMP1
4418  022604  104001                            ERROR   1
4419
4420  022606  104411                            SKIPT                       ;ERROR FATAL. GO TO NEXT TEST.
4421
4422  022610  052737  000001  177746  KX1:      BIS     #BIT0,@#CONTRL  ;DISABLE 'WARNING' TRAPS.
4423  022616  012737  022646  000114            MOV     #KXERR,@#CACHVEC        ;SET UP FOR ERROR WHICH
4424                                                                        ;SHOULD NOT TRAP!
4425  022624  012704  000400                    MOV     #400,R4             ;PATTERN FOR MAINT REG.
4426  022630  012702  177750                    MOV     #MAINT,R2
4427  022634  000240                            NOP
4428  022636  010412                            MOV     R4,(R2)             ;SET THE MAINT. REG.
4429  022640  005012                    KX2:    CLR     (R2)                ;THE FETCH OF THIS
4430  022642  000240                            NOP                         ;INSTRUCTION SHOULD CAUSE
4431  022644  000420                            BR      KXDONE              ;A CACHE  MEMORY
4432                                                                        ;PARITY ERROR WHICH
4433                                                                        ;NORMALLY SHOULD TRAP
4434                                                                        ;BUT HERE NO TRAP SHOULD
4435                                                                        ;OCCUR FOR TRAPS HAVE BEEN DISABLED.
4436
4437  022646                            KXERR:                            ;A TRAP HAS ERRONEOUSLY
4438  022646  012637  001236                    MOV     (SP)+,$TMP2         ;TAKEN PLACE, REPORT
4439  022652  005726                            TST     (SP)+               ;UNABLE TO DISABLE TRAPS.
4440  022654  013737  177746  001240            MOV     @#CONTRL,$TMP3
4441  022662  013737  177740  001242            MOV     @#LOADRS,$TMP4
4442  022670  013737  177742  001244            MOV     @#HIADRS,$TMP5
4443  022676  013737  177744  001246            MOV     @#MEMERR,$TMP6
4444
4445  022704  104134                    1$:     ERROR   134
4446
4447  022706  104410                    KXDONE: RSET
4448
4449
4450                                    ;;**************************************************************
```

```
                                        ;*TEST 46       CACHE CONTROL REGISTER DISABLE TRAPS TEST 3
                                        ;*
                                        ;*THIS IS A TEST OF THE CONTROL REGISTER'S DISABLE TRAPS FUNCTION.
                                        ;*IT IS ATTEMPTED TO DISABLE A TRAP RESULTING FROM A CACHE
                                        ;*MEMORY PARITY ERROR. THE MAINTENANCE REGISTER WILL BE USED TO
                                        ;*FORCE THE ERROR ON THE LOW BYTE OF THE , IN THE  MEMORY
                                        ;*OF GROUP 0.
                                        ;*
                                        ;*****************************************************************
022710  000004                  TST46:  SCOPE
022712  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATION
        000046                  KZ=$TN-1
                                                                ;SET THE SKAD REGISTER
022720  012737  023110  030646          MOV     #TST47,SKAD     ;IN CASE THE TEST ABORTS.

022726  113737  001102  001232          MOVB    $TSTNM,$TMP0

022734  104415                          SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
022736  104416                          SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
022740  104417                          SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
022742  104420                          SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
022744  012737  000030  177746          MOV     #SDM1,@#CONTRL  ;USE GROUP ZERO
022752  012700  023040                  MOV     #KZ2,R0         ;MAKE KZ2 A HIT IN GROUP
022756  005710                          TST     (R0)            ;ZERO.
022760  005710                          TST     (R0)

                                                                ;SEE IF REFERENCE ADDRESS
022762  032737  000010  177752          BIT     #10,@#HITMIS    ;IS A HIT.
022770  001007                          BNE     KZ1
                                                                ;IF NOT ERROR!
022772  010037  001236                  MOV     R0,$TMP2
022776  012737  000000  001234          MOV     #0,$TMP1
023004  104001                          ERROR   1

023006  104411                          SKIPT                   ;ERROR FATAL. GO TO NEXT TEST.

023010  052737  000001  177746  KZ1:    BIS     #BIT0,@#CONTRL  ;DISABLE 'WARNING' TRAPS.
023016  012737  023046  000114          MOV     #KZERR,@#CACHVEC        ;SET UP FOR ERROR WHICH
                                                                ;SHOULD NOT TRAP!
023024  012704  000020                  MOV     #20,R4          ;PATTERN FOR MAINT REG.
023030  012702  177750                  MOV     #MAINT,R2
023034  000240                          NOP
023036  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
023040  005012                  KZ2:    CLR     (R2)            ;THE FETCH OF THIS
023042  000240                          NOP                     ;INSTRUCTION SHOULD CAUSE
023044  000420                          BR      KZDONE          ;A CACHE  MEMORY
                                                                ;PARITY ERROR WHICH
                                                                ;NORMALLY SHOULD TRAP
                                                                ;BUT HERE NO TRAP SHOULD
                                                                ;OCCUR FOR TRAPS HAVE BEEN DISABLED.

023046                          KZERR:                          ;A TRAP HAS ERRONEOUSLY
023046  012637  001236                  MOV     (SP)+,$TMP2     ;TAKEN PLACE, REPORT
023052  005726                          TST     (SP)+           ;UNABLE TO DISABLE TRAPS.
023054  013737  177746  001240          MOV     @#CONTRL,$TMP3
023062  013737  177740  001242          MOV     @#LOADRS,$TMP4
```

```
 4507  023070  013737  177742  001244          MOV     @#HIADRS,$TMP5
 4508  023076  013737  177744  001246          MOV     @#MEMERR,$TMP6
 4509
 4510  023104  104134                  1$:     ERROR   134
 4511
 4512  023106  104410                  KZDONE: RSET
 4513
 4514
 4515
 4516
 4517
 4518                                  ;;*****************************************************************
 4519                                  ;*TEST 47       CACHE ERROR REGISTER LOCK UP TEST 1
 4520                                  ;*
 4521                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
 4522                                  ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
 4523                                  ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
 4524                                  ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
 4525                                  ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
 4526                                  ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
 4527                                  ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
 4528                                  ;*TO THE CACHE DIRECTLY.
 4529                                  ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
 4530                                  ;*TO THE CACHE DIRECTLY.
 4531                                  ;*
 4532                                  ;;*****************************************************************
 4533  023110  000004                  TST47:  SCOPE
 4534  023112  012737  000040  001302          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 4535          000047                  NA=$TN-1
 4536                                                                  ;SET THE SKAD REGISTER
 4537  023120  012737  023474  030646          MOV     #TST50,SKAD     ;IN CASE THE TEST ABORTS.
 4538
 4539  023126  113737  001102  001232          MOVB    $TSTNM,$TMP0
 4540
 4541  023134  104415                          SKPBER          ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
 4542  023136  104416                          SKPBCN          ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
 4543  023140  104417                          SKPBMN          ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
 4544  023142  104420                          SKPBHM          ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
 4545  023144  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
 4546
 4547
 4548  023152  012737  023226  000114          MOV     #NA3,@#CACHVEC          ;SET UP FOR THE ERROR.
 4549  023160  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
 4550  023164  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
 4551  023170  000401                          BR      NA1
 4552
 4553          023172                          LOC=.           ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 4554          023170                          LOC=-4&LOC
 4555          023174                          LOC=LOC+4
 4556          023174                          .=LOC
 4557
 4558  023174  000240                  NA1:    NOP
 4559  023176  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
 4560  023200  005701                  NA2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
 4561  023202  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
 4562  023204  000240                          NOP
```

```
4563                                                                    ;IF NONE OCCURS REPORT
4564  023206  012737  010000  001236          MOV     #10000,$TMP2      ;ERROR!
4565  023214  104127                  1$:     ERROR   127
4566  023216  012737  177777  031102          MOV     #-1,MANFL2
4567  023224  000522                  BR      NADONE
4568
4569
4570  023226                          NA3:
4571
4572  023226  012737  023302  000114          MOV     #NA6,@#CACHVEC          ;SET UP FOR THE ERROR.
4573  023234  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
4574  023240  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
4575  023244  000401                          BR      NA4
4576
4577          023246                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4578          023244                          LOC=-4&LOC
4579          023250                          LOC=LOC+4
4580          023250                          .=LOC
4581
4582  023250  000240                  NA4:    NOP
4583  023252  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
4584  023254  005701                  NA5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
4585  023256  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
4586  023260  000240                          NOP
4587                                                                  ;IF NONE OCCURS REPORT
4588  023262  012737  010000  001236          MOV     #10000,$TMP2    ;ERROR!
4589  023270  104127                  1$:     ERROR   127
4590  023272  012737  177777  031102          MOV     #-1,MANFL2
4591  023300  000474                  BR      NADONE
4592
4593
4594  023302                          NA6:
4595
4596  023302  062706  000010                  ADD     #10,SP          ;RESET THE STACK.
4597  023306  022737  144404  177744          CMP     #144404,@#MEMERR        ;SEE IF THE ERROR REGISTER
4598  023314  001004                          BNE     NA7             ;IS SET CORRECTLY.
4599  023316  022737  023200  177740          CMP     #NA2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
4600  023324  001422                          BEQ     NA8             ;IS SET CORRECTLY.
4601
4602  023326                          NA7:                            ;NOT SET CORRECTLY!
4603  023326  012737  144404  001236          MOV     #144404,$TMP2   ;REPORT FAILURE.
4604  023334  013737  177744  001240          MOV     @#MEMERR,$TMP3
4605  023342  012737  023200  001242          MOV     #NA2,$TMP4
4606  023350  005037  001244                  CLR     $TMP5
4607  023354  013737  177740  001246          MOV     @#LOADRS,$TMP6
4608  023362  013737  177742  001250          MOV     @#HIADRS,$TMP7
4609
4610  023370  104135                  1$:     ERROR   135
4611
4612  023372  005037  177572          NA8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
4613  023376  005037  172516                  CLR     @#MMR3
4614  023402  012737  177777  177744          MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
4615  023410  005737  177744                  TST     @#MEMERR        ;ERROR REG.
4616  023414  001416                          BEQ     NA10
4617
4618  023416                          NA9:                            ;WON'T CLEAR!
```

```
 4619  023416  013737  177740  001236          MOV     @#LOADRS,$TMP2
 4620  023424  013737  177742  001240          MOV     @#HIADRS,$TMP3
 4621  023432  013737  177744  001242          MOV     @#MEMERR,$TMP4
 4622
 4623  023440  104130                  1$:     ERROR   130
 4624  023442  012737  177777  031062          MOV     #-1,MMRFLG
 4625  023450  000410                          BR      NADONE
 4626
 4627  023452  022737  177740  177740  NA10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
 4628  023460  001356                          BNE     NA9              ;HAS RESET
 4629  023462  022737  000003  177742          CMP     #3,@#HIADRS
 4630  023470  001352                          BNE     NA9
 4631
 4632  023472  104410                  NADONE: RSET
 4633
 4634
 4635                                          ;;*****************************************************************
 4636                                          ;*TEST 50       CACHE ERROR REGISTER LOCK UP TEST 2
 4637                                          ;*
 4638                                          ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
 4639                                          ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
 4640                                          ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
 4641                                          ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
 4642                                          ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
 4643                                          ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
 4644                                          ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
 4645                                          ;*TO THE CACHE DIRECTLY.
 4646                                          ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
 4647                                          ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
 4648                                          ;*
 4649                                          ;;*****************************************************************
 4650  023474  000004                  TST50:  SCOPE
 4651  023476  012737  000040  001302          MOV     #40,$TIMES       ;;DO 40 ITERATIONS
 4652          000050                  NB=$TN-1
 4653                                                                   ;SET THE SKAD REGISTER
 4654  023504  012737  024164  030646          MOV     #TST51,SKAD      ;IN CASE THE TEST ABORTS.
 4655
 4656  023512  113737  001102  001232          MOVB    $TSTNM,$TMP0
 4657
 4658  023520  104415                          SKPBER                   ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
 4659  023522  104416                          SKPBCN                   ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
 4660  023524  104417                          SKPBMN                   ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
 4661  023526  104420                          SKPBHM                   ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
 4662  023530  104412                          MMSKIP
 4663
 4664  023532  012700  172340                  MOV     #KIPAR0,R0       ;SET UP MEMORY MANAGEMENT
 4665                                                                   ;TO RELOCATE EVERYTHING
 4666  023536  012702  172300                  MOV     #KIPDR0,R2       ;THROUGH THE UNIBUS
 4667  023542  012703  000007                  MOV     #7,R3            ;MAP PASSIVELY TO MEMORY,
 4668  023546  005004                          CLR     R4               ;BY PASSIVELY IS MEANT
 4669  023550  012705  170200                  MOV     #MAPL00,R5       ;THAT ADDRESS ARE
 4670                                                                   ;RELOCATED TO THEMSELVES.
 4671  023554  012722  077406          64$:    MOV     #77406,(R2)+
 4672  023560  010401                          MOV     R4,R1
 4673  023562  072127  000006                  ASH     #6,R1
 4674  023566  010125                          MOV     R1,(R5)+
```

```
4675  023570  005025                          CLR     (R5)+
4676  023572  010410                          MOV     R4,(R0)
4677  023574  062720  170000                  ADD     #170000,(R0)+
4678  023600  062704  000200                  ADD     #200,R4
4679  023604  077315                          SOB     R3,64$
4680  023606  012710  177600                  MOV     #177600,(R0)
4681  023612  012712  077406                  MOV     #77406,(R2)
4682
4683  023616  012737  000014  177746          MOV     #MDM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
4684
4685
4686  023624  012737  023702  000114          MOV     #NB3,@#CACHVEC          ;SET UP FOR THE ERROR.
4687  023632  012704  010000                  MOV     #10000,R4               ;PATTERN TO BE PUT IN
4688  023636  012702  177750                  MOV     #MAINT,R2               ;THE MAINT. REG.
4689  023642  000402                          BR      NB1
4690
4691          023644                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4692          023644                          LOC=-4&LOC
4693          023650                          LOC=LOC+4
4694          023650                          .=LOC
4695
4696  023650  000240                  NB1:    NOP
4697  023652  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
4698  023654  005701                  NB2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
4699  023656  005012                          CLR     (R2)            ;SHOULD CAUSE AN ABORT!
4700  023660  000240                          NOP
4701                                                                  ;IF NONE OCCURS REPORT
4702  023662  012737  010000  001236          MOV     #10000,$TMP2    ;ERROR!
4703  023670  104127                  1$:     ERROR   127
4704  023672  012737  177777  031102          MOV     #-1,MANFL2
4705  023700  000530                  BR      NBDONE
4706
4707
4708  023702                          NB3:
4709
4710  023702  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
4711  023710  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
4712  023716  012737  023772  000114          MOV     #NB6,@#CACHVEC  ;SET UP FOR ERROR
4713  023724  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
4714  023730  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
4715  023734  000401                          BR      NB4
4716
4717          023736                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
4718          023734                          LOC=-4&LOC
4719          023740                          LOC=LOC+4
4720          023740                          .=LOC
4721
4722  023740  000240                  NB4:    NOP
4723  023742  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
4724  023744  005701                  NB5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
4725  023746  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
4726  023750  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
4727                                                                  ;NO ABORT OCCURRED!
4728  023752  012737  010000  001236          MOV     #10000,$TMP2    ;REPORT FAILURE
4729  023760  104127                  1$:     ERROR   127
4730  023762  012737  177777  031066          MOV     #-1,MANFLG
```

```
4731  023770  000474                          BR      NBDONE
4732
4733
4734  023772                          NB6:
4735
4736  023772  062706  000010                  ADD     #10,SP          ;RESET THE STACK.
4737  023776  022737  137404  177744          CMP     #137404,@#MEMERR        ;SEE IF THE ERROR REGISTER
4738  024004  001004                          BNE     NB7             ;IS SET CORRECTLY.
4739  024006  022737  023654  177740          CMP     #NB2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
4740  024014  001422                          BEQ     NB8             ;IS SET CORRECTLY.
4741
4742  024016                          NB7:                            ;NOT SET CORRECTLY!
4743  024016  012737  137404  001236          MOV     #137404,$TMP2   ;REPORT FAILURE.
4744  024024  013737  177744  001240          MOV     @#MEMERR,$TMP3
4745  024032  012737  023654  001242          MOV     #NB2,$TMP4
4746  024040  005037  001244                  CLR     $TMP5
4747  024044  013737  177740  001246          MOV     @#LOADRS,$TMP6
4748  024052  013737  177742  001250          MOV     @#HIADRS,$TMP7
4749
4750  024060  104135                  1$:     ERROR   135
4751
4752  024062  005037  177572          NB8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
4753  024066  005037  172516                  CLR     @#MMR3
4754  024072  012737  177777  177744          MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
4755  024100  005737  177744                  TST     @#MEMERR        ;ERROR REG.
4756  024104  001416                          BEQ     NB10
4757
4758  024106                          NB9:                            ;WON'T CLEAR!
4759  024106  013737  177740  001236          MOV     @#LOADRS,$TMP2
4760  024114  013737  177742  001240          MOV     @#HIADRS,$TMP3
4761  024122  013737  177744  001242          MOV     @#MEMERR,$TMP4
4762
4763  024130  104130                  1$:     ERROR   130
4764  024132  012737  177777  031062          MOV     #-1,MMRFLG
4765  024140  000410                          BR      NBDONE
4766
4767  024142  022737  177740  177740  NB10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
4768  024150  001356                          BNE     NB9             ;HAS RESET
4769  024152  022737  000003  177742          CMP     #3,@#HIADRS
4770  024160  001352                          BNE     NB9
4771
4772  024162  104410                  NBDONE: RSET
4773
4774
4775                                  ;;*****************************************************************
4776                                  ;*TEST 51        CACHE ERROR REGISTER LOCK UP TEST 3
4777                                  ;*
4778                                  ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
4779                                  ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
4780                                  ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
4781                                  ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
4782                                  ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
4783                                  ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
4784                                  ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
4785                                  ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
4786                                  ;*THE SECOND ERROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
```

```
                                        ;*TO THE CACHE DIRECTLY.
                                        ;*
                                        ;;*****************************************************************
024164  000004                        TST51:  SCOPE
024166  012737  000040  001302              MOV     #40,$TIMES      ;;DO 40 ITERATIONS
        000051                  NC=$TN-1
                                                                    ;SET THE SKAD REGISTER
024174  012737  024664  030646              MOV     #TST52,SKAD     ;IN CASE THE TEST ABORTS.

024202  113737  001102  001232              MOVB    $TSTNM,$TMP0

024210  104415                              SKPBER                  ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
024212  104416                              SKPBCN                  ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
024214  104417                              SKPBMN                  ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
024216  104420                              SKPBHM                  ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
024220  104412                              MMSKIP

024222  012700  172340                      MOV     #KIPAR0,R0      ;SET UP MEMORY MANAGEMENT
                                                                    ;TO RELOCATE EVERYTHING
024226  012702  172300                      MOV     #KIPDR0,R2      ;THROUGH THE UNIBUS
024232  012703  000007                      MOV     #7,R3           ;MAP PASSIVELY TO MEMORY,
024236  005004                              CLR     R4              ;BY PASSIVELY IS MEANT
024240  012705  170200                      MOV     #MAPLOD,R5      ;THAT ADDRESS ARE
                                                                    ;RELOCATED TO THEMSELVES.
024244  012722  077406              64$:    MOV     #77406,(R2)+
024250  010401                              MOV     R4,R1
024252  072127  000006                      ASH     #6,R1
024256  010125                              MOV     R1,(R5)+
024260  005025                              CLR     (R5)+
024262  010410                              MOV     R4,(R0)
024264  062720  170000                      ADD     #170000,(R0)+
024270  062704  000200                      ADD     #200,R4
024274  077315                              SOB     R3,64$
024276  012710  177600                      MOV     #177600,(R0)
024302  012712  077406                      MOV     #77406,(R2)

024306  012737  000014  177746              MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.

024314  012737  000060  172516              MOV     #60,@#MMR3      ;TURN ON THE MAP AND
024322  012737  000001  177572              MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
024330  012737  024406  000114              MOV     #NC3,@#CACHVEC  ;SET UP FOR ERROR
024336  012704  010000                      MOV     #10000,R4       ;PATTERN TO BE PUT IN
024342  012702  177750                      MOV     #MAINT,R2       ;THE MAINT. REG.
024346  000402                              BR      NC1

                024350                  LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                024350                  LOC=-4&LOC
                024354                  LOC=LOC+4
                024354                  .=LOC

024354  000240                  NC1:    NOP
024356  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
024360  005701                  NC2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
024362  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
024364  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
```

```
                                                                        ;NO ABORT OCCURRED!
024366  012737  010000  001236          MOV     #10000,$TMP2          ;REPORT FAILURE
024374  104127                  1$:     ERROR   127
024376  012737  177777  031066          MOV     #-1,MANFLG
024404  000526                          BR      NCDONE

024406  005037  177572          NC3:    CLR     @#MMR0                ;TURN OFF MEMORY MANAGEMENT.
024412  005037  172516                  CLR     @#MMR3

024416  012737  024472  000114          MOV     #NC6,@#CACHVEC        ;SET UP FOR THE ERROR.
024424  012704  010000                  MOV     #10000,R4             ;PATTERN TO BE PUT IN
024430  012702  177750                  MOV     #MAINT,R2             ;THE MAINT. REG.
024434  000401                          BR      NC4

        024436                          LOC=.                 ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
        024434                          LOC=-4&LOC
        024440                          LOC=LOC+4
        024440                          .=LOC

024440  000240          NC4:    NOP
024442  010412                  MOV     R4,(R2)               ;SET THE MAINT. REG.
024444  005701          NC5:    TST     R1                    ;THE FETCH OF THIS INSTRUCTION
024446  005012                  CLR     (R2)                  ;SHOULD CAUSE AN ABORT!
024450  000240                  NOP
                                                              ;IF NONE OCCURS REPORT
024452  012737  010000  001236          MOV     #10000,$TMP2  ;ERROR!
024460  104127                  1$:     ERROR   127
024462  012737  177777  031102          MOV     #-1,MANFL2
024470  000474                  BR      NCDONE

024472          NC6:

024472  062706  000010                  ADD     #10,SP                ;RESET THE STACK.
024476  022737  167404  177744          CMP     #167404,@#MEMERR              ;SEE IF THE ERROR REGISTER
024504  001004                          BNE     NC7                   ;IS SET CORRECTLY.
024506  022737  024360  177740          CMP     #NC2,@#LOADRS         ;SEE IF THE ADDRESS REGISTER
024514  001422                          BEQ     NC8                   ;IS SET CORRECTLY.

024516          NC7:                                                  ;NOT SET CORRECTLY!
024516  012737  167404  001236          MOV     #167404,$TMP2         ;REPORT FAILURE.
024524  013737  177744  001240          MOV     @#MEMERR,$TMP3
024532  012737  024360  001242          MOV     #NC2,$TMP4
024540  005037  001244                  CLR     $TMP5
024544  013737  177740  001246          MOV     @#LOADRS,$TMP6
024552  013737  177742  001250          MOV     @#HIADRS,$TMP7

024560  104135                  1$:     ERROR   135

024562  005037  177572          NC8:    CLR     @#MMR0                ;TURN OFF MEMORY MANAGEMENT.
024566  005037  172516                  CLR     @#MMR3
024572  012737  177777  177744          MOV     #-1,@#MEMERR          ;SEE IF YOU CAN CLR THE
024600  005737  177744                  TST     @#MEMERR              ;ERROR REG.
024604  001416                          BEQ     NC10
```

```
 4899  024606                                 NC9:                                    ;WON'T CLEAR!
 4900  024606  013737  177740  001236                MOV     @#LOADRS,$TMP2
 4901  024614  013737  177742  001240                MOV     @#HIADRS,$TMP3
 4902  024622  013737  177744  001242                MOV     @#MEMERR,$TMP4
 4903
 4904  024630  104130                         1$:    ERROR   130
 4905  024632  012737  177777  031062                MOV     #-1,MMRFLG
 4906  024640  000410                                BR      NCDONE
 4907
 4908  024642  022737  177740  177740  NC10:         CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
 4909  024650  001356                                BNE     NC9              ;HAS RESET
 4910  024652  022737  000003  177742                CMP     #3,@#HIADRS
 4911  024660  001352                                BNE     NC9
 4912
 4913  024662  104410                         NCDONE: RSET
 4914
 4915
 4916                                         ;;**********************************************************************
 4917                                         ;*TEST 52       CACHE ERROR REGISTER LOCK UP TEST 4
 4918                                         ;*
 4919                                         ;*THIS IS A TEST OF THE ERROR REGISTER'S ABILITY TO LOCK UP ON
 4920                                         ;*THE FIRST ERROR WHEN A SERIES OF ERRORS OCCUR. ALSO TESTED IS THE
 4921                                         ;*ERROR ADDRESS'S ABILITY TO LOCK ON THE ADDRESS OF THE FIRST
 4922                                         ;*ERROR IN A SEQUENCE OF ERRORS. IN THIS TEST TWO ERROR ARE FORCED
 4923                                         ;*ON TOP OF EACH OTHER, BOTH OF THEM WILL BE ERRORS TO
 4924                                         ;*THE MAIN MEMORY WANTED WORD DATA PARITY ERRORS! THE FIRST
 4925                                         ;*REFERENCE RESULTING IN AN ERROR WILL BE MADE FROM THE CPU
 4926                                         ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
 4927                                         ;*THE SECOND EPROR WILL BE BECAUSE OF A REFERENCE FROM THE CPU
 4928                                         ;*TO THE UNIBUS THROUGH THE MAPPING BOX TO THE CACHE.
 4929                                         ;*
 4930                                         ;;**********************************************************************
 4931  024664  000004                         TST52:  SCOPE
 4932  024666  012737  000040  001302                MOV     #40,$TIMES       ;;DO 40 ITERATIONS
 4933          000052                         ND=$TN-1
 4934                                                                          ;SET THE SKAD REGISTER
 4935  024674  012737  025370  030646                MOV     #TST53,SKAD      ;IN CASE THE TEST ABORTS.
 4936
 4937  024702  113737  001102  001232                MOVB    $TSTNM,$TMP0
 4938
 4939  024710  104415                                SKPBER                   ;IF THE ERROR REGISTER IS BAD SKIP THIS TEST.
 4940  024712  104416                                SKPBCN                   ;IF THE CONTROL REGISTER IS BAD SKIP THIS TEST.
 4941  024714  104417                                SKPBMN                   ;IF THE MAINTENANCE REGISER IS BAD SKIP TEST.
 4942  024716  104420                                SKPBHM                   ;IF THE HIT/MISS REGISTER IS BAD SKIP THIS TEST.
 4943  024720  104412                                MMSKIP
 4944
 4945  024722  012700  172340                        MOV     #KIPAR0,R0       ;SET UP MEMORY MANAGEMENT
 4946                                                                          ;TO RELOCATE EVERYTHING
 4947  024726  012702  172300                        MOV     #KIPDR0,R2       ;THROUGH THE UNIBUS
 4948  024732  012703  000007                        MOV     #7,R3            ;MAP PASSIVELY TO MEMORY,
 4949  024736  005004                                CLR     R4               ;BY PASSIVELY IS MEANT
 4950  024740  012705  170200                        MOV     #MAPLOO,R5       ;THAT ADDRESS ARE
 4951                                                                          ;RELOCATED TO THEMSELVES.
 4952  024744  012722  077406                64$:    MOV     #77406,(R2)+
 4953  024750  010401                                MOV     R4,R1
 4954  024752  072127  000006                        ASH     #6,R1
```

```
 4955  024756  010125                          MOV     R1,(R5)+
 4956  024760  005025                          CLR     (R5)+
 4957  024762  010410                          MOV     R4,(R0)
 4958  024764  062720  170000                  ADD     #170000,(R0)+
 4959  024770  062704  000200                  ADD     #200,R4
 4960  024774  077315                          SOB     R3,64$
 4961  024776  012710  177600                  MOV     #177600,(R0)
 4962  025002  012712  077406                  MOV     #77406,(R2)
 4963
 4964  025006  012737  000014  177746          MOV     #MOM1,@#CONTRL          ;FORCE MISSES TO BOTH GROUPS.
 4965
 4966
 4967  025014  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
 4968  025022  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
 4969  025030  012737  025106  000114          MOV     #ND3,@#CACHVEC  ;SET UP FOR ERROR
 4970  025036  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
 4971  025042  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
 4972  025046  000402                          BR      ND1
 4973
 4974          025050                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 4975          025050                          LOC=-4&LOC
 4976          025054                          LOC=LOC+4
 4977          025054                          .=LOC
 4978
 4979  025054  000240                  ND1:    NOP
 4980  025056  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
 4981  025060  005701                  ND2:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
 4982  025062  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
 4983  025064  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
 4984                                                                  ;NO ABORT OCCURRED!
 4985  025066  012737  010000  001236          MOV     #10000,$TMP2    ;REPORT FAILURE
 4986  025074  104127                  1$:     ERROR   127
 4987  025076  012737  177777  031066          MOV     #-1,MANFLG
 4988  025104  000530                          BR      NDDONE
 4989
 4990
 4991  025106                          ND3:
 4992
 4993  025106  012737  000060  172516          MOV     #60,@#MMR3      ;TURN ON THE MAP AND
 4994  025114  012737  000001  177572          MOV     #1,@#MMR0       ;22-BIT MODE ADDRESSING
 4995  025122  012737  025176  000114          MOV     #ND6,@#CACHVEC  ;SET UP FOR ERROR
 4996  025130  012704  010000                  MOV     #10000,R4       ;PATTERN TO BE PUT IN
 4997  025134  012702  177750                  MOV     #MAINT,R2       ;THE MAINT. REG.
 4998  025140  000401                          BR      ND4
 4999
 5000          025142                          LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
 5001          025140                          LOC=-4&LOC
 5002          025144                          LOC=LOC+4
 5003          025144                          .=LOC
 5004
 5005  025144  000240                  ND4:    NOP
 5006  025146  010412                          MOV     R4,(R2)         ;SET THE MAINT. REG.
 5007  025150  005701                  ND5:    TST     R1              ;THE FETCH OF THIS INSTRUCTION
 5008  025152  005012                          CLR     (R2)            ;SHOULD CASE AN ABORT
 5009  025154  000240                          NOP                     ;AND UNIBUS PB ASSERTED!
 5010                                                                  ;NO ABORT OCCURRED!
```

```
5011  025156  012737  010000  001236          MOV     #10000,$TMP2    ;REPORT FAILURE
5012  025164  104127                  1$:     ERROR   127
5013  025166  012737  177777  031066          MOV     #-1,MANFLG
5014  025174  000474                          BR      ND0ONE
5015
5016
5017  025176                          ND6:
5018
5019  025176  062706  000010                  ADD     #10,SP          ;RESET THE STACK.
5020  025202  022737  033404  177744          CMP     #33404,@#MEMERR ;SEE IF THE ERROR REGISTER
5021  025210  001004                          BNE     ND7             ;IS SET CORRECTLY.
5022  025212  022737  025060  177740          CMP     #ND2,@#LOADRS   ;SEE IF THE ADDRESS REGISTER
5023  025220  001422                          BEQ     ND8             ;IS SET CORRECTLY.
5024
5025  025222                          ND7:                            ;NOT SET CORRECTLY!
5026  025222  012737  033404  001236          MOV     #33404,$TMP2    ;REPORT FAILURE.
5027  025230  013737  177744  001240          MOV     @#MEMERR,$TMP3
5028  025236  012737  025060  001242          MOV     #ND2,$TMP4
5029  025244  005037  001244                  CLR     $TMP5
5030  025250  013737  177740  001246          MOV     @#LOADRS,$TMP6
5031  025256  013737  177742  001250          MOV     @#HIADRS,$TMP7
5032
5033  025264  104135                  1$:     ERROR   135
5034
5035  025266  005037  177572          ND8:    CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT.
5036  025272  005037  172516                  CLR     @#MMR3
5037  025276  012737  177777  177744          MOV     #-1,@#MEMERR    ;SEE IF YOU CAN CLR THE
5038  025304  005737  177744                  TST     @#MEMERR        ;ERROR REG.
5039  025310  001416                          BEQ     ND10
5040
5041  025312                          ND9:                            ;WON'T CLEAR!
5042  025312  013737  177740  001236          MOV     @#LOADRS,$TMP2
5043  025320  013737  177742  001240          MOV     @#HIADRS,$TMP3
5044  025326  013737  177744  001242          MOV     @#MEMERR,$TMP4
5045
5046  025334  104130                  1$:     ERROR   130
5047  025336  012737  177777  031062          MOV     #-1,MMRFLG
5048  025344  000410                          BR      ND0ONE
5049
5050  025346  022737  177740  177740  ND10:   CMP     #177740,@#LOADRS ;SEE IF THE ADDRESS REGSTER
5051  025354  001356                          BNE     ND9             ;HAS RESET
5052  025356  022737  000003  177742          CMP     #3,@#HIADRS
5053  025364  001352                          BNE     ND9
5054
5055  025366  104410                  ND0ONE: RSET
5056
5057
5058                                  ;;*****************************************************************
5059                                  ;*TEST 53        MAIN MEMORY DATA PARITY CHECKERS LOW BYTE TEST
5060                                  ;*
5061                                  ;*THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS
5062                                  ;*FOR THE LOW BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD.
5063                                  ;*THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY
5064                                  ;*ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY
5065                                  ;*BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE
5066                                  ;*THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT
```

```
                                        ;*A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE
                                        ;*AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY
                                        ;*BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS
                                        ;*SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1).
                                        ;*THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA
                                        ;*PARITY CHECKERS WORKS IN SUCH A WAY AS TO
                                        ;*EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO
                                        ;*THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO
                                        ;*AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS
                                        ;*ALREADY ONE THEN NO ERROR OCCURS!
                                        ;*
                                        ;;*****************************************************************
025370  000004                  TST53:  SCOPE
025372  012737  000020  001302          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000054                  UA=$TN
                                                                ;SET THE SKAD REGISTER
025400  012737  025744  030646          MOV     #TST54,SKAD     ;IN CASE THE TEST ABORTS.

025406  113737  001102  001232          MOVB    $TSTNM,$TMPO
025414  012737  030522  000114          MOV     #SPUR,@#CACHVEC

025422  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
025430  005000                          CLR     R0              ;INITIALIZE

025432  012737  025432  001110  UA1:    MOV     #UA1,$LPERR
025440  004737  031106                  JSR     PC,PARCNT       ;SEE IF THE CURRENT TEST
025444  032702  000001                  BIT     #BIT0,R2        ;PATTERN HAS THE PARITY BIT
025450  001002                          BNE     UA2             ;OFF, IF NOT GO TO NEXT
025452  000137  025724                  JMP     UA7             ;PATTERN

025456  012737  025630  000114  UA2:    MOV     #UAER1,@#CACHVEC        ;SET UP FOR THE ERROR, EVEN WORD.
025464  012704  010000                  MOV     #10000,R4       ;THIS IS A PATTERN WHICH
025470  012702  177750                  MOV     #MAINT,R2       ;WHEN LOADED INTO THE
                                                                ;MAINTENANCE REGISTER
                                                                ;WILL FORCE AN ERROR ON
                                                                ;THE MAIN MEMORY EVEN
025474  012701  025624                  MOV     #UATMP1,R1      ;WORD LOW BYTE
025500  010011                          MOV     R0,(R1)
025502  010412                          MOV     R4,(R2)         ;SET THE MAINT REG
025504  021101                          CMP     (R1),R1         ;THE REFERENCE TO (R1),
                                                                ;UATMP1 SHOULD CAUSE
                                                                ;AN ERROR.
025506  005012                          CLR     (R2)
025510  005012                          CLR     (R2)

025512                          UA3:
                                                                ;THE ERROR DIDN'T OCCUR!
025512  010037  001236                  MOV     R0,$TMP2        ;REPORT FAILURE
025516  012737  025624  001240          MOV     #UATMP1,$TMP3
025524  005037  001242                  CLR     $TMP4
025530  104140                  64$:    ERROR   140

025532  012737  025670  000114  UA4:    MOV     #UAER2,@#CACHVEC        ;SET UP FOR THE ERROR
025540  012737  025532  001110          MOV     #UA4,$LPERR     ;ON THE ODD WORD.
025546  012704  040000                  MOV     #40000,R4       ;THIS IS A PATTERN WHICH
025552  012702  177750                  MOV     #MAINT,R2       ;WHEN LOADED IN THE MAINTENANCE
```

```
                                                                        ;REGISTER WILL CAUSE AN ERROR
025556  012701  025626                  MOV     #UATMP2,R1            ;ON THE ODD WORD, LOW BYTE.
025562  010011                          MOV     R0,(R1)               ;SET THE MAINT REG. AND
025564  000240                          NOP
025566  010412                          MOV     R4,(R2)               ;REFERENCE (R1), UATMP2, AND
025570  021101                          CMP     (R1),R1               ;CAUSE THE ERROR.

025572  005012                          CLR     (R2)
025574  005012                          CLR     (R2)

025576                          UA5:
                                                                      ;THE ERROR DIDN'T OCCUR!
025576  010037  001236                  MOV     R0,$TMP2              ;REPORT FAILURE
025602  012737  025626  001240          MOV     #UATMP2,$TMP3
025610  005037  001242                  CLR     $TMP4
025614  104141                  64$:    ERROR   141

025616  000442                  UA6:    BR      UA7


                025620                  LOC=.                 ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
                025620                  LOC=-4&LOC
                025624                  LOC=LOC+4
                025624                  .=LOC

025624  000000                  UATMP1:.WORD    0
025626  000000                  UATMP2:.WORD    0

025630                          UAER1:
025630  022737  104404  177744          CMP     #104404,@#MEMERR        ;MAKE SURE THE ERROR
025636  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
025640  000137  030522          1$:     JMP     SPUR
025644  022737  025624  177740  2$:     CMP     #UATMP1,@#LOADRS        ;MAKE SURE THE ERROR
025652  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
025654  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
025656  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
025664  000137  025532                  JMP     UA4             ;GO TEST THE ODD WORD

025670                          UAER2:
025670  022737  104410  177744          CMP     #104410,@#MEMERR        ;MAKE SURE THE ERROR
025676  001402                          BEQ     2$              ;REGISTER IS SET PROPERLY
025700  000137  030522          1$:     JMP     SPUR
025704  022737  025626  177740  2$:     CMP     #UATMP2,@#LOADRS        ;MAKE SURE THE ERROR
025712  001372                          BNE     1$              ;OCCURRED AT THE CORRECT
                                                                ;ADDRESS.
025714  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
025716  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.

025724  022700  000377          UA7:    CMP     #377,R0         ;INCREMENT THE TEST PATTERN
025730  001404                          BEQ     UA8
025732  062700  000001                  ADD     #1,R0
025736  000137  025432                  JMP     UA1

025742  104410                  UA8:    RSET
```

```
                        ;;*****************************************************************
                        ;*TEST 54       MAIN MEMORY DATA PARITY CHECKERS HIGH BYTE TEST
                        ;*
                        ;*THIS IS A TEST OF THE TWO MAIN MEMORY DATA PARITY CHECKERS
                        ;*FOR THE HIGH BYTE, ONE FOR EACH OF THE EVEN AND ODD WORD.
                        ;*THE MAINTENANCE REGISTER IS USED TO FORCE A PARITY
                        ;*ERROR AT EVERY DATA PATTERN, WHICH HAS A ZERO PARITY
                        ;*BIT, THAT CAN BE WRITTEN INTO AN 8-BIT BYTE. NOTE
                        ;*THAT MAIN MEMORY HAS ODD PARITY WHICH MEANS THAT
                        ;*A BYTE WILL HAVE A ZERO PARITY BIT IF THERE ARE
                        ;*AN ODD NUMBER OF BITS SET (1) IN THAT BYTE. THE PARITY
                        ;*BIT WOULD BE ONE (SET) FOR A BYTE WHICH HAD NO BITS
                        ;*SET (1) OR A BYTE WHICH HAD AN EVEN NUMBER OF BITS SET (1).
                        ;*THE MAINTENANCE FUNCTION FOR THE MAIN MEMORY DATA
                        ;*PARITY CHECKERS WORKS IN SUCH A WAY AS TO
                        ;*EFFECTIVELY FORCE THE BYTES PARITY BIT TO ONE (SET), SO
                        ;*THAT IF THE PARITY BIT FOR THAT BYTE HAD BEEN ZERO
                        ;*AN ERROR OCCURS! IF THE BYTE'S PARITY BIT WAS
                        ;*ALREADY ONE THEN NO ERROR OCCURS!
                        ;*
                        ;;*****************************************************************
025744  000004         TST54:  SCOPE
025746  012737  000020  001302          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
        000055         UB=$TN
                                                                ;SET THE SKAD REGISTER
025754  012737  026320  030646          MOV     #TST55,SKAD     ;IN CASE THE TEST ABORTS.

025762  113737  001102  001232          MOVB    $TSTNM,$TMPO
025770  012737  030522  000114          MOV     #SPUR,@#CACHVEC

025776  012737  000014  177746          MOV     #MOM1,@#CONTRL  ;FORCE MISSES TO BOTH GROUPS.
026004  005000                         CLR     R0              ;INITIALIZE

026006  012737  026006  001110  UB1:    MOV     #UB1,$LPERR
026014  004737  031106                 JSR     PC,PARCNT       ;SEE IF THE CURRENT TEST
026020  032702  000001                 BIT     #BIT0,R2        ;PATTERN HAS THE PARITY BIT
026024  001002                         BNE     UB2             ;OFF, IF NOT GO TO NEXT
026026  000137  026300                 JMP     UB7             ;PATTERN

026032  012737  026204  000114  UB2:    MOV     #UBER1,@#CACHVEC        ;SET UP FOR THE ERROR, EVEN WORD.
026040  012704  020000                 MOV     #20000,R4       ;THIS IS A PATTERN WHICH
026044  012702  177750                 MOV     #MAINT,R2       ;WHEN LOADED INTO THE
                                                               ;MAINTENANCE REGISTER
                                                               ;WILL FORCE AN ERROR ON
                                                               ;THE MAIN MEMORY EVEN
026050  012701  026200                 MOV     #UBTMP1,R1      ;WORD HIGH BYTE
026054  010011                         MOV     R0,(R1)
026056  010412                         MOV     R4,(R2)         ;SET THE MAINT REG
026060  021101                         CMP     (R1),R1         ;THE REFERENCE TO (R1),
                                                               ;UBTMP1 SHOULD CAUSE
                                                               ;AN ERROR.
026062  005012                         CLR     (R2)
026064  005012                         CLR     (R2)

026066                         UB3:
                                                               ;THE ERROR DIDN'T OCCUR!
```

```
5235  026066  010037  001236                    MOV     R0,$TMP2        ;REPORT FAILURE
5236  026072  012737  026200  001240            MOV     #UBTMP1,$TMP3
5237  026100  005037  001242                    CLR     $TMP4
5238  026104  104142                  64$:      ERROR   142
5239
5240  026106  012737  026244  000114  UB4:      MOV     #UBER2,@#CACHVEC        ;SET UP FOR THE ERROR
5241  026114  012737  026106  001110            MOV     #UB4,$LPERR     ;ON THE ODD WORD.
5242  026122  012704  100000                    MOV     #100000,R4      ;THIS IS A PATTERN WHICH
5243  026126  012702  177750                    MOV     #MAINT,R2       ;WHEN LOADED IN THE MAINTENANCE
5244                                                                    ;REGISTER WILL CAUSE AN ERROR
5245  026132  012701  026202                    MOV     #UBTMP2,R1      ;ON THE ODD WORD, LOW BYTE.
5246  026136  010011                            MOV     R0,(R1)         ;SET THE MAINT REG. AND
5247  026140  000240                            NOP
5248  026142  010412                            MOV     R4,(R2)         ;REFERENCE (R1), UBTMP2, AND
5249  026144  021101                            CMP     (R1),R1         ;CAUSE THE ERROR.
5250
5251  026146  005012                            CLR     (R2)
5252  026150  005012                            CLR     (R2)
5253
5254  026152                           UB5:
5255                                                                    ;THE ERROR DIDN'T OCCUR!
5256  026152  010037  001236                    MOV     R0,$TMP2        ;REPORT FAILURE
5257  026156  012737  026202  001240            MOV     #UBTMP2,$TMP3
5258  026164  005037  001242                    CLR     $TMP4
5259  026170  104143                  64$:      ERROR   143
5260
5261  026172  000442                  UB6:      BR      UB7
5262
5263
5264          026174                            LOC=.                   ;GET THE PC TO AN EVEN WORD BOUNDARY!!!
5265          026174                            LOC=-4&LOC
5266          026200                            LOC=LOC+4
5267          026200                            .=LOC
5268
5269  026200  000000                  UBTMP1:.WORD    0
5270  026202  000000                  UBTMP2:.WORD    0
5271
5272  026204                           UBER1:
5273  026204  022737  104404  177744            CMP     #104404,@#MEMERR        ;MAKE SURE THE ERROR
5274  026212  001402                            BEQ     2$              ;REGISTER IS SET PROPERLY
5275  026214  000137  030522          1$:       JMP     SPUR
5276  026220  022737  026200  177740  2$:       CMP     #UBTMP1,@#LOADRS        ;MAKE SURE THE ERROR
5277  026226  001372                            BNE     1$              ;OCCURRED AT THE CORRECT
5278                                                                    ;ADDRESS.
5279  026230  022626                            CMP     (SP)+,(SP)+     ;RESET THE STACK
5280  026232  012737  177777  177744            MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
5281  026240  000137  026106                    JMP     UB4             ;GO TEST THE ODD WORD
5282
5283  026244                           UBER2:
5284  026244  022737  104410  177744            CMP     #104410,@#MEMERR        ;MAKE SURE THE ERROR
5285  026252  001402                            BEQ     2$              ;REGISTER IS SET PROPERLY
5286  026254  000137  030522          1$:       JMP     SPUR
5287  026260  022737  026202  177740  2$:       CMP     #UBTMP2,@#LOADRS        ;MAKE SURE THE ERROR
5288  026266  001372                            BNE     1$              ;OCCURRED AT THE CORRECT
5289                                                                    ;ADDRESS.
5290  026270  022626                            CMP     (SP)+,(SP)+     ;RESET THE STACK
```

```
5291  026272  012737  177777  177744          MOV     #-1,@#MEMERR    ;CLEAR THE ERROR REGISTERS.
5292
5293  026300  022700  177400          UB7:    CMP     #177400,R0              ;INCREMENT THE TEST PATTERN
5294  026304  001404                          BEQ     UB8
5295  026306  062700  000400                  ADD     #400,R0
5296  026312  000137  026006                  JMP     UB1
5297
5298  026316  104410                  UB8:    RSET
5299
5300  026320                          TST55:
5301
5302
5303                                  .SBTTL  END OF PASS ROUTINE
5304
5305                                  ;;**********************************************************************
5306                                  ;*INCREMENT THE PASS NUMBER ($PASS)
5307                                  ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
5308                                  ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
5309                                  ;*IF THERES A MONITOR GO TO IT
5310                                  ;*IF THERE ISN'T JUMP TO LOOP
5311
5312  026320                          $EOP:
5313  026320  000004                          SCOPE
5314  026322  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
5315  026326  005037  001302                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
5316  026332  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
5317  026336  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
5318  026344  005327                          DEC     (PC)+           ;;LOOP?
5319  026346  000001                  $EOPCT: .WORD   1
5320  026350  003031                          BGT     $DOAGN          ;;YES
5321  026352  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
5322  026354  000001                  $ENDCT: .WORD   1
5323  026356  026346                          $EOPCT
5324  026360  104401  026443                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
5325  026364  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
5326  026370  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
5327  026372  104401  026440                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
5328  026376  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
5329  026402  001414                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
5330  026404  012703  125252                  MOV     #125252,R3
5331  026410  004737  031156                  JSR     PC,CHAINQ
5332  026414  013700  000042                  MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
5333  026420  001405                          BEQ     $DOAGN          ;;RETURN ADDRESS
5334  026422  000005                          RESET                   ;;CLEAR THE WORLD
5335  026424  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
5336  026426  000240                          NOP                     ;;SAVE ROOM
5337  026430  000240                          NOP                     ;;FOR
5338  026432  000240                          NOP                     ;;ACT11
5339  026434                          $DOAGN:
5340  026434  000137                          JMP     @(PC)+          ;;RETURN
5341  026436  003364                  $RTNAD: .WORD   LOOP
5342  026440     377      377      000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
5343  026443     015  042412  042116 $ENDMG: .ASCIZ  <15><12>/END PASS #/
5344  026450  050040  051501  020123
5345  026456  000043
5346
```

```
                                        .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

026460                                  $SCOPE:
026460  032777  040000  152452  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
026466  001114                         BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
026470  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
026472  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
026476  012737  026516  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
026504  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
026510  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
026514  000466                         BR      $SVLAD          ;;GO TO THE NEXT TEST
026516  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
026520  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
026524  000426                         BR      7$              ;;LOOP ON THE PRESENT TEST
026526                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
026526  032777  000400  152404          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
026534  001407                         BEQ     2$              ;;BR IF NO
026536  017746  152376                  MOV     @SWR,-(SP)      ;;SET DESIRED TEST NUM. FROM SWR
026542  042716  000200                  BIC     #$SWRMK,(SP)    ;;STRIP AWAY UNDESIRED BITS
026546  122637  001102                  CMPB    (SP)+,$TSTNM    ;;ON THE RIGHT TEST?
026552  001462                         BEQ     $OVER           ;;BR IF YES
026554  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
026560  001421                         BEQ     3$              ;;BR IF NO
026562  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
026570  101015                         BHI     3$              ;;BR IF NO
026572  032777  001000  152340          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
026600  001404                         BEQ     4$              ;;BR IF NO
026602  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
026610  000443                         BR      $OVER
026612  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
026616  005037  001302                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
026622  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
026624  032777  004000  152306  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
026632  001011                         BNE     1$              ;;BR IF YES
026634  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
026640  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
026642  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
026646  023737  001302  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
026654  002021                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
026656  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
026664  013737  026734  001302          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
026672  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
```

```
 5403  026676  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 5404  026702  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 5405  026706  005037  001304                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 5406  026712  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 5407  026720  013777  001102  152214  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 5408  026726  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 5409  026732  000002                          RTI                     ;;FIXES PS
 5410  026734  000001                  $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
 5411
 5412                                  .SBTTL  ERROR HANDLER ROUTINE
 5413
 5414                                  ;;**********************************************************************
 5415                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 5416                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 5417                                  ;*AND GO TO ERTYPE ON ERROR
 5418                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 5419                                  ;*SW15=1         HALT ON ERROR
 5420                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
 5421                                  ;*SW10=1         BELL ON ERROR
 5422                                  ;*SW09=1         LOOP ON ERROR
 5423                                  ;*CALL
 5424                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 5425
 5426  026736                          $ERROR:
 5427  026736  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 5428  026742  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 5429  026744  013777  001102  152170          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 5430  026752  032777  002000  152160          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 5431  026760  001402                          BEQ     1$              ;;NO - SKIP
 5432  026762  104401  001306                  TYPE    ,$BELL          ;;RING BELL
 5433  026766  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 5434  026772  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 5435  026776  162737  000002  001116          SUB     #2,$ERRPC
 5436  027004  117737  152106  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 5437  027012  032777  020000  152120          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 5438  027020  001004                          BNE     20$             ;;SKIP TYPEOUTS
 5439  027022  004737  031352                  JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
 5440  027026  104401  001313                  TYPE    ,$CRLF
 5441  027032                          20$:
 5442  027032  005777  152102          2$:     TST     @SWR            ;;HALT ON ERROR
 5443  027036  100001                          BPL     3$              ;;SKIP IF CONTINUE
 5444  027040  000000                          HALT                    ;;HALT ON ERROR!
 5445  027042  032777  001000  152070  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 5446  027050  001402                          BEQ     4$              ;;BR IF NO
 5447  027052  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 5448  027056  005737  001304          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 5449  027062  001402                          BEQ     5$              ;;BR IF NONE
 5450  027064  013716  001304                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 5451  027070                          5$:
 5452  027070  022737  026424  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 5453  027076  001001                          BNE     6$              ;;BRANCH IF NO
 5454  027100  000000                          HALT                    ;;YES
 5455  027102                          6$:
 5456  027102  012737  177777  177744          MOV     #-1,@#MEMERR
 5457  027110  005037  177766                  CLR     @#CPUERR
 5458  027114  000002                          RTI
```

```
                                        .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                        ;;**************************************************************************
                                        ;*SAVE R0-R5
                                        ;*CALL:
                                        ;*      SAVREG
                                        ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                        ;*
                                        ;*TOP---(+16)
                                        ;* +2---(+18)
                                        ;* +4---R5
                                        ;* +6---R4
                                        ;* +8---R3
                                        ;*+10---R2
                                        ;*+12---R1
                                        ;*+14---R0

027116                                  $SAVREG:
027116  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
027120  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
027122  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
027124  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
027126  010446                                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
027130  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
027132  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
027136  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
027142  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
027146  016646  000022                          MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
027152  000002                                  RTI

                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
027154                                  $RESREG:
027154  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
027160  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
027164  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
027170  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
027174  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
027176  012604                                  MOV     (SP)+,R4        ;;POP STACK INTO R4
027200  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
027202  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
027204  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
027206  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
027210  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
                                        ;*

027212  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
027216  100002                          BPL     1$              ;;BR IF YES
027220  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
027222  000407                          BR      3$              ;;LEAVE
027224  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
027226  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
027232  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
027234  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
027236  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
027240  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
027242  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
027246  000002                          RTI                     ;;RETURN
027250  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
027254  001430                          BEQ     8$
027256  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
027262  001006                          BNE     5$
027264  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
027266  104401                          TYPE                    ;;TYPE A CR AND LF
027270  001313                          $CRLF
027272  105037  027426                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
027276  000755                          BR      2$              ;;GET NEXT CHARACTER
027300  004737  027362          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
027304  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
027310  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
027312  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
027316  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
027322  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
027324  004737  027362                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
027330  105337  027426                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
027334  000770                          BR      7$              ;;LOOP

                                        ;HORIZONTAL TAB PROCESSOR

027336  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
027342  004737  027362          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
027346  132737  000007  027426          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
027354  001372                          BNE     9$              ;;TAB STOP
027356  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
027360  000724                          BR      2$              ;;GET NEXT CHARACTER
027362  105777  151562          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
027366  100375                          BPL     $TYPEC
027370  116677  000002  151554          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
027376  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
027404  001003                          BNE     1$              ;;BRANCH IF NO
027406  105037  027426                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
027412  000406                          BR      $TYPEX          ;;EXIT
027414  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
```

```
                                        BEQ     $TYPEX          ;;BRANCH IF YES
                                        INCB    (PC)+           ;;COUNT THE CHARACTER
                                $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
                                $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

 027432  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 027436  116637  000001 027655         MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 027444  112637  027657                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 027450  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 027454  000406                        BR      $TYPON
 027456  112737  000001 027655 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 027464  112737  000006 027657         MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 027472  112737  000005 027654 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 027500  010346                        MOV     R3,-(SP)        ;;SAVE R3
 027502  010446                        MOV     R4,-(SP)        ;;SAVE R4
 027504  010546                        MOV     R5,-(SP)        ;;SAVE R5
 027506  113704  027657                MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 027512  005404                        NEG     R4
 027514  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 027520  110437  027656                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 027524  113704  027655                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 027530  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 027534  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
 027536  006105                 1$:    ROL     R5              ;;ROTATE MSB INTO "C"
 027540  000404                        BR      3$              ;;GO DO MSB
 027542  006105                 2$:    ROL     R5              ;;FORM THIS DIGIT
 027544  006105                        ROL     R5
 027546  006105                        ROL     R5
 027550  010503                        MOV     R5,R3
 027552  006103                 3$:    ROL     R3              ;;GET LSB OF THIS DIGIT
```

```
 5627   027554   105337   027656                  DECB    $OMODE          ;;TYPE THIS DIGIT?
 5628   027560   100016                           BPL     7$              ;;BR IF NO
 5629   027562   042703   177770                  BIC     #177770,R3      ;;GET RID OF JUNK
 5630   027566   001002                           BNE     4$              ;;TEST FOR 0
 5631   027570   005704                           TST     R4              ;;SUPPRESS THIS 0?
 5632   027572   001403                           BEQ     5$              ;;BR IF YES
 5633   027574   005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 5634   027576   052703   000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 5635   027602   052703   000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 5636   027606   110337   027652                  MOVB    R3,8$           ;;SAVE FOR TYPING
 5637   027612   104401   027652                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
 5638   027616   105337   027654          7$:     DECB    $OCNT           ;;COUNT BY 1
 5639   027622   003347                           BGT     2$              ;;BR IF MORE TO DO
 5640   027624   002402                           BLT     6$              ;;BR IF DONE
 5641   027626   005204                           INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 5642   027630   000744                           BR      2$              ;;GO DO THE LAST DIGIT
 5643   027632   012605                   6$:     MOV     (SP)+,R5        ;;RESTORE R5
 5644   027634   012604                           MOV     (SP)+,R4        ;;RESTORE R4
 5645   027636   012603                           MOV     (SP)+,R3        ;;RESTORE R3
 5646   027640   016666   000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 5647   027646   012616                           MOV     (SP)+,(SP)
 5648   027650   000002                           RTI                     ;;RETURN
 5649   027652      000                   8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 5650   027653      000                           .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 5651   027654      000                   $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 5652   027655      000                   $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 5653   027656   000000                   $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 5654
 5655                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 5656
 5657                                    ;;**************************************************************
 5658                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 5659                                    ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 5660                                    ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 5661                                    ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 5662                                    ;*REPLACED WITH SPACES.
 5663                                    ;*CALL:
 5664                                    ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 5665                                    ;*      TYPDS                   ;;GO TO THE ROUTINE
 5666
 5667   027660                           $TYPDS:
 5668   027660   010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 5669   027662   010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5670   027664   010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 5671   027666   010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 5672   027670   010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 5673   027672   012746   020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 5674   027676   016605   000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 5675   027702   100004                           BPL     1$              ;;BR IF INPUT IS POS.
 5676   027704   005405                           NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 5677   027706   112766   000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 5678   027714   005000                   1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 5679   027716   012703   030074                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 5680   027722   112723   000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 5681   027726   005002                   2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 5682   027730   016001   030064                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
```

```
5683   027734  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
5684   027736  002402                          BLT     4$              ;;BR IF DONE
5685   027740  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
5686   027742  000774                          BR      2$
5687   027744  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
5688   027746  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
5689   027750  001002                          BNE     5$              ;;FALL THROUGH IF 0
5690   027752  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
5691   027754  100407                          BMI     7$              ;;BR IF YES
5692   027756  106316                  5$:     ASLB    (SP)            ;;MSD?
5693   027760  103003                          BCC     6$              ;;BR IF NO
5694   027762  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
5695   027770  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
5696   027774  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
5697   030000  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
5698   030002  005720                          TST     (R0)+           ;;JUST INCREMENTING
5699   030004  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
5700   030010  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
5701   030012  003002                          BGT     8$              ;;GO TO EXIT
5702   030014  010502                          MOV     R5,R2           ;;GET THE LSD
5703   030016  000764                          BR      6$              ;;GO CHANGE TO ASCII
5704   030020  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
5705   030022  100003                          BPL     9$              ;;BR IF NO
5706   030024  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
5707   030032  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
5708   030034  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
5709   030036  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
5710   030040  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
5711   030042  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
5712   030044  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5713   030046  104401  030074                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
5714   030052  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
5715   030060  012616                          MOV     (SP)+,(SP)
5716   030062  000002                          RTI                     ;;RETURN TO USER
5717   030064  023420                  $DTBL:  10000.
5718   030066  001750                          1000.
5719   030070  000144                          100.
5720   030072  000012                          10.
5721   030074  000004                  $DBLK:  .BLKW   4
5722
5723                                    .SBTTL  TRAP DECODER
5724
5725                                    ;;**************************************************************
5726                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
5727                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5728                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5729                                    ;*GO TO THAT ROUTINE.
5730
5731   030104  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
5732   030106  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
5733   030112  005740                          TST     -(R0)           ;;BACKUP BY 2
5734   030114  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
5735   030116  006300                          ASL     R0              ;;POSITION FOR INDEXING
5736   030120  016000  030140                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
5737   030124  000200                          RTS     R0              ;;GO TO ROUTINE
5738
```

```
                                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
                                        $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
                                                MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
                                                RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
                                        $TRPAD: .WORD   $TRAP2
                                                $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
                                                $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
                                                $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
                                                $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
                                                $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

                                                $SAVREG ;;CALL=SAVREG   TRAP+6(104406)  SAVE R0-R5 ROUTINE
                                                $RESREG ;;CALL=RESREG   TRAP+7(104407)  RESTORE R0-R5 ROUTINE

                                                CLEAN   ;;CALL=RSET     TRAP+10(104410) GO RESET ALL REGISTERS.
                                                ABORTT  ;;CALL=SKIPT    TRAP+11(104411) THIS WILL SKIP TO THE NEXT TEST
                                                MMDES   ;;CALL=MMSKIP   TRAP+12(104412) IF SWITCH # IS ON SKIP TO THE NEXT TEST
                                                MSIZER  ;;CALL=SIZE     TRAP+13(104413) DETERMINE THE HIGHEST ADDRESS IN MEMORY
                                                SKBADR  ;;CALL=SKPBAD   TRAP+14(104414) SKIP TEST IF ERROR ADDRESS REGISTER IS I
                                                SKBERR  ;;CALL=SKPBER   TRAP+15(104415) SKIP TEST IF ERROR REGISTER IS INOPERATI
                                                SKBCNR  ;;CALL=SKPBCN   TRAP+16(104416) SKIP TEST IF CONTROL REGISTER IS INOPERA
                                                SKBMNR  ;;CALL=SKPBMN   TRAP+17(104417) SKIP TEST IF MAINTENANCE REGISTER IS INO
                                                SKBHMR  ;;CALL=SKPBHM   TRAP+20(104420) SKIP TEST IF HIT/MISS REGISTER IS IN OPE

                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;**************************************************************
                                        ;POWER DOWN ROUTINE
                                        $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
                                                MOV     #340,@#PWRVEC+2 ;;PRIO:7
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                                MOV     R4,-(SP)        ;;PUSH R4 ON STACK
                                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
                                                MOV     SP,$SAVR6       ;;SAVE SP
                                                MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
                                                HALT
                                                BR      .-2             ;;HANG UP

                                        ;;**************************************************************
                                        ;POWER UP ROUTINE
                                        $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
```

```
5795  030262  013706  030352                MOV     $SAVR6,SP       ;;GET SP
5796  030266  005037  030352                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
5797  030272  005237  030352          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
5798  030276  001375                       BNE     1$              ;;OF  WORD
5799  030300  012677  150634                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
5800  030304  012605                       MOV     (SP)+,R5        ;;POP STACK INTO R5
5801  030306  012604                       MOV     (SP)+,R4        ;;POP STACK INTO R4
5802  030310  012603                       MOV     (SP)+,R3        ;;POP STACK INTO R3
5803  030312  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
5804  030314  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
5805  030316  012600                       MOV     (SP)+,R0        ;;POP STACK INTO R0
5806  030320  012737  030202  000024       MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
5807  030326  012737  000340  000026       MOV     #340,@#PWRVEC+2 ;;PRIO:7
5808  030334  104401                       TYPE                    ;;REPORT THE POWER FAILURE
5809  030336  032127                $PWRMG: .WORD   POWERM          ;;POWER FAIL MESSAGE POINTER
5810  030340  012716                       MOV     (PC)+,(SP)      ;;RESTART AT START
5811  030342  003016                $PWRAD: .WORD   START           ;;RESTART ADDRESS
5812  030344  000002                       RTI
5813  030346  000000                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
5814  030350  000776                       BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
5815  030352  000000                $SAVR6: 0                       ;;PUT THE SP HERE
5816
5817                                .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
5818
5819                                ;;**************************************************************
5820                                ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
5821                                ;*UNSIGNED OCTAL ASCIZ NUMBER.
5822                                ;*CALL
5823                                ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
5824                                ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
5825                                ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
5826
5827
5828  030354  104406                $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
5829  030356  016601  000002               MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
5830  030362  012705  030473               MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
5831  030366  012704  000014               MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
5832  030372  012703  177770               MOV     #^C7,R3         ;;MASK
5833  030376  012100                       MOV     (R1)+,R0        ;;LOWER WORD
5834  030400  012101                       MOV     (R1)+,R1        ;;HIGH WORD
5835  030402  005002                       CLR     R2              ;;TERMINATOR
5836  030404  110245                1$:    MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
5837  030406  010002                       MOV     R0,R2           ;;GET THIS DIGIT
5838  030410  005304                       DEC     R4              ;;COUNT THIS CHARACTER
5839  030412  003007                       BGT     3$              ;;BR IF NOT THE LAST DIGIT
5840  030414  001405                       BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
5841  030416  005205                       INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
5842  030420  010566  000002               MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
5843  030424  104407                       RESREG                  ;;RESTORE ALL REGISTERS
5844  030426  000207                       RTS     PC              ;;RETURN TO USER
5845  030430  006203                2$:    ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
5846  030432  006001                3$:    ROR     R1              ;;POSITION THE BINARY NUMBER FOR
5847  030434  006000                       ROR     R0              ;;      THE NEXT OCTAL DIGIT
5848  030436  006001                       ROR     R1
5849  030440  006000                       ROR     R0
5850  030442  006001                       ROR     R1
```

```
 5851  030444  006000                          ROR     R0
 5852  030446  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
 5853  030450  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
 5854  030454  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
 5855  030456  000016                $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
 5856
 5857                                  ;THIS ROUTINE IS CALLED BY UNEXPECTED TRAPS TO VECTOR ERRVEC.
 5858                                  ;THE ERROR IS REPORTED AND CONTROL IS TRANSFERRED BACK TO THE TEST
 5859                                  ;FOLLOWING THE ONE THAT WAS INTERRUPTED WHEN THE ERROR OCCURRED!
 5860  030474  011637  001234          CPSPUR: MOV     (SP),$TMP1
 5861  030500  012737  030516  001236          MOV     #1$,$TMP2
 5862  030506  013737  177766  001240          MOV     @#CPUERR,$TMP3
 5863  030514  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK
 5864  030516  104150                  1$:     ERROR   150
 5865  030520  104411                          SKIPT
 5866
 5867                                  ;THIS ROUTINE HANDLE UNEXPECTED TRAPS TO #CACHVEC.
 5868  030522  012737  030612  000114  $PUR:   MOV     #10$,@#CACHVEC
 5869  030530  013700  177744                  MOV     @#MEMERR,R0
 5870  030534  032700  000014                  BIT     #14,R0          ;SEE IF IT WAS A MAIN MEMORY PARITY ERROR.
 5871  030540  001403                          BEQ     9$
 5872  030542  013700  177740                  MOV     @#LOADRS,R0     ;IF IT WAS THEN THE BAD PARITY IS
 5873  030546  005710                          TST     (R0)            ;CACHED AND MUST BE PURGED!!!!!
 5874  030550  012737  030522  000114  9$:     MOV     #$PUR,@#CACHVEC
 5875  030556  013737  177744  001242          MOV     @#MEMERR,$TMP4  ;TRAP HERE IF AN UNEXPECTED
 5876  030564  013737  177740  001234          MOV     @#LOADRS,$TMP1  ;ERROR, PARITY, OCCURS.
 5877  030572  013737  177742  001236          MOV     @#HIADRS,$TMP2
 5878  030600  011637  001240                  MOV     (SP),$TMP3
 5879  030604  022626                          CMP     (SP)+,(SP)+
 5880  030606  104014                  1$:     ERROR   14
 5881  030610  104411                          SKIPT                   ;?????
 5882  030612  022626                  10$:    CMP     (SP)+,(SP)+
 5883  030614  000137  030550                  JMP     9$
 5884
 5885                                  ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL SKIPT.
 5886                                  ;IT TELLS THE USER THAT THE CURRENT TEST HAS BEEN
 5887                                  ;ABORTED AND THAT CONTROL IS BEING PASSED TO THE NEXT TEST.
 5888  030620  011637  001234  ABORTT: MOV     (SP),$TMP1
 5889  030624  112737  000015  001114          MOVB    #15,$ITEMB
 5890  030632  022626                          CMP     (SP)+,(SP)+
 5891  030634  004737  031352                  JSR     PC,ERTYPE
 5892  030640  104410                          RSET
 5893  030642  000177  000000                  JMP     @SKAD           ;GO TO @SKAD, WHICH SHOULD
 5894                                                                  ;BE SET TO THE
 5895  030646  000000                  SKAD:   .WORD   0               ;ADDRESS OF THE NEXT TEST.
 5896
 5897
 5898                                  ;THIS ROUTINE IS CALLED BY THE TRAP CATCHER CALL RSET. IT CLEARS ALL
 5899                                  ;THE IMPORTANE REGISTERS AND RESETS THE STACK.
 5900  030650                          CLEAN:
 5901
 5902  030650  012737  030522  000114          MOV     #$PUR,@#CACHVEC
 5903  030656  012737  030474  000004          MOV     #CPSPUR,@#ERRVEC
 5904  030664  011637  030736                  MOV     (SP),BACKAD
 5905  030670  012706  001100                  MOV     #STACK,SP
 5906  030674  005037  177750                  CLR     @#MAINT         ;CLEAR ALL CONTROL AND ERROR
```

```
5907  030700  005037  177572                  CLR     @#MMR0          ;REGISTERS.
5908  030704  005037  172516                  CLR     @#MMR3
5909  030710  005037  177746                  CLR     @#CONTRL
5910  030714  012737  177777  177744          MOV     #-1,@#MEMERR
5911  030722  005037  177766                  CLR     @#CPUERR
5912  030726  005037  177776                  CLR     @#PSW
5913  030732  000177  000000                  JMP     @BACKAD
5914  030736  000000                BACKAD: .WORD   0
5915
5916
5917                                ;COME HERE TO TEST THE REGISTER FLAGS AND USE THEM TO DETERMINE WHETHER
5918                                ;OR NOT TO SKIP A TEST WHICH RELIES ON THE FUNCTIONALLITY OF THAT REGISTER
5919                                ;TO BE PROPERLY RUN.
5920                                ;THESE ROUTINES ARE CALLED BY THE TRAP CATCHER CALLS:
5921                                ;       SKPBAD          SKIPT IF BAD ERROR ADDRESS REGISTER
5922                                ;       SKPBER          SKIPT IF BAD ERROR REGISTER
5923                                ;       SKPBCN          SKIPT IF BAD CONTROL REGISTER
5924                                ;       SKPBMN          SKIPT IF BAD MAINTENANCE REGISTER
5925                                ;       SKPBHM          SKIPT IF BAD HIT/MISS REGISTER
5926                                ;
5927
5928  030740  005737  031056        SKBADR: TST     LOAFLG
5929  030744  001004                        BNE     1$
5930  030746  005737  031060                TST     HIAFLG
5931  030752  001001                        BNE     1$
5932  030754  000002                        RTI
5933  030756  104401                1$:     TYPE
5934  030760  033111                        .WORD   ADRNG
5935  030762  000433                        BR      SKRNG
5936
5937  030764  005737  031062        SKBERR: TST     MMRFLG
5938  030770  001001                        BNE     1$
5939  030772  000002                        RTI
5940  030774  104401                1$:     TYPE
5941  030776  033221                        .WORD   ERRNG
5942  031000  000424                        BR      SKRNG
5943
5944  031002  005737  031064        SKBCNR: TST     CONFLG
5945  031006  001001                        BNE     1$
5946  031010  000002                        RTI
5947  031012  104401                1$:     TYPE
5948  031014  033321                        .WORD   CNRNG
5949  031016  000415                        BR      SKRNG
5950
5951  031020  005737  031066        SKBMNR: TST     MANFLG
5952  031024  001001                        BNE     1$
5953  031026  000002                        RTI
5954  031030  104401                1$:     TYPE
5955  031032  033423                        .WORD   MNRNG
5956  031034  000406                        BR      SKRNG
5957
5958  031036  005737  031070        SKBHMR: TST     HIMFLG
5959  031042  001001                        BNE     1$
5960  031044  000002                        RTI
5961  031046  104401                1$:     TYPE
5962  031050  033531                        .WORD   HMRNG
```

```
 5963
 5964  031052  022626                SKRNG:  CMP     (SP)+,(SP)+             ;RESET THE STACK AND GO TO THE
 5965  031054  104411                        SKIPT                           ;NEXT TEST!!!!!
 5966
 5967  031056  000000                LOAFLG: .WORD   0                       ;THESE ARE FLAGS USED TO DESIGNATE
 5968  031060  000000                HIAFLG: .WORD   0                       ;EITHER A GOOD OR A BAD REGISTER.
 5969  031062  000000                MMRFLG: .WORD   0                       ;GOOD WILL BE DESIGNATED BY A
 5970  031064  000000                CONFLG: .WORD   0                       ;0 BAD BY A NOT ZERO!!
 5971  031066  000000                MANFLG: .WORD   0
 5972  031070  000000                HIMFLG: .WORD   0
 5973  031072  000000                LOAFL2: .WORD   0
 5974  031074  000000                HIAFL2: .WORD   0
 5975  031076  000000                MMRFL2: .WORD   0
 5976  031100  000000                CONFL2: .WORD   0
 5977  031102  000000                MANFL2: .WORD   0
 5978  031104  000000                HIMFL2: .WORD   0
 5979
 5980                                ;THIS ROUTINE IS CALLED TO DETERMINE THE PARITY OF
 5981                                ;A DATA PATTERN. THE PATTERN WHICH IS TAKEN BY THIS
 5982                                ;ROUTINE AS ITS ARGUMENT SHOULD BE PUT IN R0. THEN
 5983                                ;TRANSFER CONTROL HERE BY EXECUTING:
 5984                                ;       JSR     PC,PARCNT
 5985                                ;WHEN THIS ROUTINE RETURNS THE NUMBER OF ON,(1), BITS
 5986                                ;IN R0 IS LEFT IN R2. THIS WOULD BE A NUMBER BETWEEN
 5987                                ;0 AND 16.
 5988  031106  012701  000001       PARCNT: MOV     #1,R1
 5989  031112  005002                        CLR     R2
 5990  031114  030100                1$:     BIT     R1,R0
 5991  031116  001401                        BEQ     2$
 5992  031120  005202                        INC     R2
 5993  031122  006301                2$:     ASL     R1
 5994  031124  103373                        BCC     1$
 5995  031126  000207                        RTS     PC
 5996
 5997                                ;THIS ROUTINE IS CALLED TO RESTORE THE TOP 1500 (DEC) WORDS IN THE
 5998                                ;FIRST 28K OF MEMORY. THIS SHOULD EFFECTIVELY RESTORE ANY MONITOR
 5999                                ;OR LOADER THAT WAS PRESENT BEFORE THIS PROGRAM BEGAN EXECUTION.
 6000                                ;CONTROL IS PASSED TO THIS ROUTINE BY AN INTERRUPT FROM THE TTY KEYBOARD
 6001                                ;WHEN ANY CHARACTER IS TYPED ON THE KEYBOARD. IF THE CHARACTER.
 6002                                ;TURNS OUT TO BE A ↑C (CONTROL-C) THEN MEMORY IS RESTORED. IF THE
 6003                                ;CHARACTER IS NOT ↑C THEN A RETURN IS MADE TO THE TEST FOLLOWING
 6004                                ;THE ONE WHOSE EXECUTION WAS INTERRUPTED BY THE KEYBOARD INTERRUPT.
 6005  031130  017700  150012        RESMON: MOV     @$TKB,R0
 6006  031134  104410                        RSET
 6007  031136  005003                        CLR     R3
 6008  031140  042700  000200              BIC     #BIT7,R0        ;GET THE CHARACTER, INITIALIZE THE REGISTERS
 6009  031144  022700  000003              CMP     #3,R0           ;AND SEE IF THE CHARACTER WAS ↑C.
 6010  031150  001027                      BNE     NOCNC           ;BRANCH AND GO TO NEXT TEST IF NOT.
 6011  031152  104401                      TYPE                    ;ECHOE THE CONTROL-C AS '↑C'
 6012  031154  032064                      .WORD   CONCMS
 6013  031156  012704  002734        CHAINQ: MOV     #↑D1500,R4              ;AND RESTORE THE MONITOR.
 6014  031162  012701  051310              MOV     #BOTTOM+4,R1
 6015  031166  012702  160000              MOV     #160000,R2
 6016  031172  012142                1$:     MOV     (R1)+,-(R2)
 6017  031174  077402                      SOB     R4,1$
 6018  031176  012737  177777  031252      MOV     #-1,MONF        ;RESET THE MONITOR RESTORED FLAG.
```

```
 6019  031204  020327  125252                    CMP     R3,#125252      ;SEE IF THE MONITOR IS BEING RESTORED
 6020                                                                      ;BY THE .$EOP ROUTINE.
 6021  031210  001001                            BNE     STOP            ;IF NOT GO HALT, OTHERWISE RETURN TO .$EOP
 6022  031212  000207                            RTS     PC
 6023  031214  104401                  STOP:     TYPE                    ;TYPE THE MONITOR RESTORED MESSAGE.
 6024  031216  032070                            .WORD   MMESRS
 6025  031220  013737  031250  000060            MOV     MONTTY,@#TKVEC
 6026  031226  000000                            HALT                    ;AND HALT!!
 6027  031230  005077  147712            NOCNC:  CLR     @$TKB           ;NOT CONTROL C SO RETURN TO NEXT TEST.
 6028  031234  152777  000100  147702            BISB    #BIT6,@$TKS
 6029  031242  104410                            RSET
 6030  031244  000177  177376                    JMP     @SKAD           ;RETURN.
 6031  031250  000000                  MONTTY:   .WORD   0               ;TEMPORARY STORAGE FOR THE INITIAL
 6032                                                                      ;CONTENTS OF THE TTY KEYBOARD INTERRUPT VECTOR.
 6033  031252  177777                  MONF:     .WORD   177777          ;FLAG, IF NOT -1 THE MONITOR IS SAVED!!
 6034
 6035
 6036                                  ;THIS ROUTINE IS CALLED BY THE TRAP CALL MMSKIP. IT LOOKS
 6037                                  ;AT THE SWITCH REGISTER AND DETERMINES WHETHER OR NOT
 6038                                  ;SWITCH #7 IS ON. IF SO THE CURRENT TEST IS SKIPPED
 6039                                  ;AND THE NEXT TEST IS ENTERED. A SSKAD MUST BE ISSUED
 6040                                  ;BEFORE THE MMSKIP.
 6041                                  ;THE PURPOSE OF SWITCH #7 IS TO CAUSE THE DELETION OF THE
 6042                                  ;EXECUTION OF ANY TEST WHICH RELIES ON MEMORY MANAGEMENT
 6043                                  ;FOR ITS OPERATION.
 6044
 6045  031254  032737  000200  001140  MMDES:   BIT     #SW7,@#SWR
 6046  031262  001001                            BNE     1$              ;IS THE SWITCH ON?
 6047  031264  000002                            RTI                     ;NO, SO RETURN.
 6048  031266  022626                  1$:       CMP     (SP)+,(SP)+
 6049  031270  104410                            RSET
 6050  031272  000177  177350                    JMP     @SKAD           ;YES, GO TO THE NEXT TEST.
 6051                                  ;THIS ROUTINE IS CALLED TO DETERMINE THE HIGHEST POSSIBLE
 6052                                  ;ADDRESS IN MEMORY. IT IS CALLED THUS, BY TRAP CALL SIZE:
 6053                                  ;               SIZE
 6054                                  ;       LOORDA: .WORD   0
 6055                                  ;       HIORDA: .WORD   0
 6056                                  ;       NXTINST:
 6057                                  ;THE LOW ORDER 16-BITS OF THE ADDRESS ARE LEFT IN THE
 6058                                  ;WORD DIRECTLY FOLLOWING THE CALL. THE HIGH ORDER 6-BITS
 6059                                  ;ARE LEFT IN THE NEXT WORD AND CONTROL IS RETURNED
 6060                                  ;TO THE THIRD WORD FOLLOWING THE CALL.
 6061  031276  010046                  MSIZER:   MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0 AND R1
 6062  031300  010146                            MOV     R1,-(SP)        ;GET THE ADDRESS OF
 6063  031302  016600  000004                    MOV     4(SP),R0        ;THE CALL OF THE STACK.
 6064  031306  013710  177760                    MOV     @#SIZELO,(R0)
 6065  031312  005060  000002                    CLR     2(R0)
 6066  031316  012701  000006                    MOV     #6,R1           ;ROTATE THE 16-BIT 'BLOCK'
 6067                                                                      ;NUMBER 6-BITS TO THE
 6068  031322  006310                  1$:       ASL     (R0)            ;LEFT AND TURN ON LOW ORDER
 6069  031324  006160  000002                    ROL     2(R0)           ;BITS 1-5 LEAVING BIT-0
 6070  031330  077104                            SOB     R1,1$           ;OFF SO AS TO CREATE
 6071  031332  052710  000076                    BIS     #76,(R0)        ;THE 22-BIT PHYSICAL ADDRESS OF
 6072                                                                      ;THE HIGHEST WORD IN
 6073                                                                      ;MEMORY.
 6074  031336  022020                            CMP     (R0)+,(R0)+     ;DETERMINE THE RETURN ADDRESS
```

```
 6075  031340  010066  000004                 MOV     R0,4(SP)        ;AND LEAVE ON THE STACK FOR
 6076                                                                   ;AN RTI.
 6077  031344  012601                         MOV     (SP)+,R1        ;RESTORE R1 AND R0.
 6078  031346  012600                         MOV     (SP)+,R0
 6079  031350  000002                         RTI                     ;RETURN
 6080                                 ;THIS ROUTINE IS USED TO TYPE AN ERROR MESSAGE
 6081                                 ;WHICH IS IN THE DATA TABLE. IT IS CALLED BY
 6082                                 ;THE $ERROR ROUTINE OR BY FIRST SETTING THE $ITEMB
 6083                                 ;BYTE EQUAL TO THE ERROR TABLE ITEM NUMBER THAT IS
 6084                                 ;TO BE PRINTED OUT AND THEN EXECUTING A JSR PC,ERTYPE
 6085  031352  104401                 ERTYPE: TYPE
 6086  031354  001313                         .WORD   $CRLF
 6087  031356  010046                         MOV     R0,-(SP)        ;SAVE R0
 6088  031360  005000                         CLR     R0
 6089  031362  113700  001114                 MOVB    $ITEMB,R0       ;GET THE ITEM NUMBER
 6090  031366  001005                         BNE     1$              ;ZERO?
 6091  031370  013746  001116                 MOV     $ERRPC,-(SP)    ;YES, TYPE JUST THE PC
 6092  031374  104402                         TYPOC                   ;OF THE ERROR CALL.
 6093  031376  000137  031714                 JMP     ERT5
 6094
 6095  031402  005300                 1$:     DEC     R0              ;MAKE R0 AN INDEX FOR THE
 6096  031404  072027  000003                 ASH     #3,R0           ;ERROR TABLE
 6097  031410  062700  001316                 ADD     #$ERRTB,R0
 6098  031414  012037  031424                 MOV     (R0)+,2$        ;TYPE EM, ERROR MESSAGE.
 6099  031420  001404                         BEQ     3$
 6100  031422  104401                         TYPE
 6101  031424  000000                 2$:     .WORD   0
 6102  031426  104401                         TYPE
 6103  031430  001313                         .WORD   $CRLF
 6104  031432  012037  031442         3$:     MOV     (R0)+,4$        ;TYPE DH, DATA HEADER
 6105  031436  001404                         BEQ     5$
 6106  031440  104401                         TYPE
 6107  031442  000000                 4$:     .WORD   0
 6108  031444  104401                         TYPE
 6109  031446  001313                         .WORD   $CRLF
 6110  031450  010146                 5$:     MOV     R1,-(SP)        ;SAVE R1
 6111  031452  012001                         MOV     (R0)+,R1        ;GET DT, DATA TABLE ADDRESS
 6112  031454  001002                         BNE     6$
 6113  031456  000137  031712                 JMP     ERT4            ;JMP IF NO ERROR TABLE.
 6114  031462  012000                 6$:     MOV     (R0)+,R0        ;GET DF, DATA FORMAT ADDRESS
 6115  031464  105710                 ERT1:   TSTB    (R0)            ;DATA FORMAT ENTRY EQUALS
 6116  031466  001003                         BNE     7$              ;ZERO?
 6117  031470  013146                         MOV     @(R1)+,-(SP)    ;YES, SO TYPE A 16-BIT
 6118  031472  104402                         TYPOC                   ;OCTAL NUMBER
 6119  031474  000500                         BR      ERT2
 6120  031476  122710  000001         7$:     CMPB    #1,(R0)         ;FORMAT EQUALS 1?
 6121  031502  001003                         BNE     8$
 6122  031504  013146                         MOV     @(R1)+,-(SP)    ;YES, TYPE A DECIMAL NUMBER
 6123  031506  104405                         TYPDS
 6124  031510  000472                         BR      ERT2
 6125
 6126  031512  122710  000002         8$:     CMPB    #2,(R0)         ;FORMAT 2?
 6127  031516  001012                         BNE     9$
 6128  031520  012146                 85$:    MOV     (R1)+,-(SP)     ;YES, TYPE A 22-BIT NUMBR
 6129  031522  004737  030354                 JSR     PC,$DB2O        ;CALL $DB2O TO CONVERT THE
 6130  031526  062716  000003                 ADD     #3,(SP)         ;BINARY TO ASCII
```

```
6131   031532   012637   031540                MOV     (SP)+,29$       ;TYPE THE STRING
6132   031536   104401                         TYPE
6133   031540   000000                 29$:    .WORD   0
6134   031542   000455                         BR      ERT2
6135
6136   031544   122710   000004         9$:     CMPB    #4,(R0)         ;FORMAT 4?
6137   031550   001004                         BNE     10$
6138   031552   013146                         MOV     @(R1)+,-(SP)    ;YES, TYPE A 16-BIT
6139   031554   104403                         TYPOS                   ;OCTAL NUMBER SUPRESSING
6140   031556      016                         .BYTE   16              ;LEADING ZEROES
6141   031557      000                         .BYTE   0
6142   031560   000446                         BR      ERT2
6143   031562   122710   000003         10$:    CMPB    #3,(R0)         ;FORMAT 3?
6144   031566   001007                         BNE     11$
6145   031570   013146                         MOV     @(R1)+,-(SP)    ;YES CONVERT 16-BIT
6146   031572   012737   177777 031720         MOV     #-1,TVADFL      ;VIRTUAL ADDRESS TO 32-BIT
6147   031600   004737   031726                JSR     PC,TYPVAD       ;PHYSICAL ADDRESS AND TYPE
6148   031604   000434                         BR      ERT2            ;RELOCATE ONLY IF SEG. IS ON!
6149   031606   122710   000005         11$:    CMPB    #5,(R0)         ;FORMAT 5?
6150   031612   001005                         BNE     12$
6151   031614   012137   031622                MOV     (R1)+,20$       ;PRINT ASCIZ STRING
6152   031620   104401                         TYPE
6153   031622   000000                 20$:    .WORD   0
6154   031624   000426                         BR      ERT3
6155
6156   031626   122710   000006         12$:    CMPB    #6,(R0)         ;FORMAT 6
6157   031632   001005                         BNE     13$
6158   031634   005037   031720                CLR     TVADFL
6159   031640   004737   031726                JSR     PC,TYPVAD
6160   031644   000414                         BR      ERT2
6161
6162   031646   122710   000007         13$:    CMPB    #7,(R0)         ;FORMAT 7?
6163   031652   001010                         BNE     14$
6164   031654   012146                         MOV     (R1)+,-(SP)
6165   031656   004737   030354                JSR     PC,$DB2O
6166   031662   012637   031670                MOV     (SP)+,45$
6167   031666   104401                         TYPE
6168   031670   000000                 45$:    .WORD   0
6169   031672   000401                         BR      ERT2
6170
6171   031674   000000                 14$:    HALT                    ;?????
6172
6173   031676   104401                 ERT2:   TYPE                    ;PRINT A TAB AFTER TYPING AN
6174   031700   032174                         .WORD   $TAB            ;ERROR TABLE ENTRY OF ALL MODES
6175                                                                   ;EXCEPT ASCIZ
6176   031702   005200                 ERT3:   INC     R0              ;POINT TO THE NEXT FORMAT BYTE
6177   031704   005711                         TST     (R1)            ;IS THERE ANOTHER ENTRY?
6178   031706   001401                         BEQ     ERT4
6179   031710   000665                         BR      ERT1            ;YES, PROCESS IT
6180                                                                   ;OTHERWISE:
6181   031712   012601                 ERT4:   MOV     (SP)+,R1        ;RESTORE R1
6182   031714   012600                 ERT5:   MOV     (SP)+,R0        ;RESTORE R0
6183   031716   000207                         RTS     PC              ;AND RETURN
6184
6185   031720   000000                 TVADFL: .WORD   0               ;FLAG USED TO TELL TYVAD
6186                                                                   ;WHETHER TO CONDITIONALLY
```

```
                                                                ;OR UNCONDITIONALLY RELOCATE
                                                                ;WHEN TYPING AN ADDRESS,
                                                                ;-1 OR 0 RESPECTIVELY

031722  000000                TVADLO: .WORD   0                   ;REGISTERS FOR THE 22-BIT
031724  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.
031726  104406              TYPVAD: SAVREG
031730  016601  000002              MOV     2(SP),R1            ;GET THE VIRTUAL
031734  010137  031722              MOV     R1,TVADLO           ;ADDRESS
031740  005037  031724              CLR     TVADHI
031744  005737  031720              TST     TVADFL              ;CONDITIONALLY RELOCATE?
031750  001404                      BEQ     1$
031752  032737  000001  177572      BIT     #1,@#MMR0           ;YES, SEE IF MEMORY
031760  001424                      BEQ     2$                  ;MANAGEMENT IS ON
031762  005000              1$:     CLR     R0                  ;RELOCATE
031764  073027  000003              ASHC    #3,R0               ;LEFT SHIFT R0 AND R1
031770  006300                      ASL     R0                  ;THREE PLACES. R0 ONE
                                                                ;MORE SO THAT IT CONTAINS
                                                                ;2 X THE UPPER 3-BITS OF
031772  000241                      CLC                         ;THE VIRTUAL ADDRESS
031774  006001                      ROR     R1                  ;RESTORE R1 TO THE OFFSET
031776  006001                      ROR     R1                  ;OF THE VIRTUAL ADDRESS
032000  006001                      ROR     R1                  ;TO THE PAR
032002  062700  172340              ADD     #KIPAR0,R0          ;DETERMINE THE CORRECT PAR'S
                                                                ;ADDRESS
032006  011003                      MOV     (R0),R3             ;GET ITS CONTENTS
032010  005002                      CLR     R2
032012  073227  000006              ASHC    #6,R2               ;MAKE THE BLOCK COUNT
                                                                ;A 22-BIT ADDRESS.
032016  060103                      ADD     R1,R3               ;ADD THE OFFSET TO THE
032020  005502                      ADC     R2                  ;BASE ADDRESS

032022  010237  031724              MOV     R2,TVADHI
032026  010337  031722              MOV     R3,TVADLO
032032  012746  031722      2$:     MOV     #TVADLO,-(SP)       ;CALL $DB2O TO CONVERT THE
032036  004737  030354              JSR     PC,$DB2O            ;22-BIT
032042  062716  000003              ADD     #3,(SP)             ;TYPE ONLY 8 DIGITS.
032046  012637  032054              MOV     (SP)+,3$
032052  104401                      TYPE
032054  000000              3$:     .WORD   0
032056  104407                      RESREG                      ;RESTORE THE REGISTERS
032060  012616                      MOV     (SP)+,(SP)          ;LEAVE ONLY THE RETURN
                                                                ;ADDRESS ON THE STACK.
032062  000207                      RTS     PC                  ;RETURN

                            ;SPECIAL MESSAGES:

032064  041536  000200      CONCMS: .ASCIZ  '↑C'<CRLF>

032070  047515  044516  047524  MMESRS: .ASCIZ  'MONITOR (OR LOADER) RESTORED!'<CRLF>
```

```
  6243  032076  020122  047450  020122
  6244  032104  047514  042101  051105
  6245  032112  020051  042522  052123
  6246  032120  051117  042105  100041
  6247  032126     000
  6248
  6249  032127     200  047520  042527  POWERM: .ASCIZ  <CRLF>'POWER FAILURE, PROGRAM RESTARTING'<CRLF><CRLF>
  6250  032134  020122  040506  046111
  6251  032142  051125  026105  050040
  6252  032150  047522  051107  046501
  6253  032156  051040  051505  040524
  6254  032164  052122  047111  100107
  6255  032172  000200
  6256
  6257  032174  000011                  $TAB:   .ASCIZ  <TAB>
  6258
  6259  032176  042600  050130  041505  MTA5:   .ASCII  <CRLF>'EXPECTED DATA:'<CRLF>
  6260  032204  042524  020104  040504
  6261  032212  040524  100072
  6262  032216  051107  052517  020120          .ASCIZ  'GROUP 0.GROUP 1.MEM EV.'<TAB>'MEM ODD.'<CRLF>
  6263  032224  027060  051107  052517
  6264  032232  020120  027061  042515
  6265  032240  020115  053105  004456
  6266  032246  042515  020115  042117
  6267  032254  027104  000200
  6268
  6269  032260  042200  052101  020101  MTA11:  .ASCII  <CRLF>'DATA WRITTEN.'<TAB>'TEST ADDR  TAB>'ERROR REG.'<CRLF>
  6270  032266  051127  052111  042524
  6271  032274  027116  052011  051505
  6272  032302  020124  042101  051104
  6273  032310  004456  051105  047522
  6274  032316  020122  042522  027107
  6275  032324     200
  6276
  6277  032325     040  047111  000040  MTA17:  .ASCIZ  ' IN '
  6278
  6279  032332  054105  042520  052103  MTB17:  .ASCIZ  'EXPECTED DATA:'<CRLF>
  6280  032340  042105  042040  052101
  6281  032346  035101  000200
  6282
  6283  032352  054502  042524  004456  MTC17:  .ASCIZ  'BYTE.'<TAB>
  6284  032360     000
  6285
  6286  032361     127  051117  027104  MTA20:  .ASCIZ  'WORD.'<TAB>
  6287  032366  000011
  6288
  6289  032370  054105  042520  052103  MTA21:  .ASCII  'EXPECTED DATA:'<CRLF>
  6290  032376  042105  042040  052101
  6291  032404  035101     200
  6292  032407     110  052111  020123          .ASCIZ  'HITS IN GROUP 0.'<TAB>'/'<TAB>'HITS IN GROUP 1. '<CRLF>
  6293  032414  047111  043440  047522
  6294  032422  050125  030040  004456
  6295  032430  004457  044510  051524
  6296  032436  044440  020116  051107
  6297  032444  052517  020120  027061
  6298  032452  100040     000
```

```
6299
6300                032325                          MTB21=MTA17
6301
6302   032455       200   042524   052123  MTA43:  .ASCII  <CRLF>'TEST ADDRESS.'<TAB>'ERROR ADRS REG.'<TAB>
6303   032462    040440   042104   042522
6304   032470    051523   004456   051105
6305   032476    047522   020122   042101
6306   032504    051522   051040   043505
6307   032512    004456
6308   032514    051105   047522   020122          .ASCIZ  'ERROR REG.'<CRLF>
6309   032522    042522   027107   000200
6310
6311   032530    053600   047522   042524  MTA45:  .ASCIZ  <CRLF>'WROTE. 377'<TAB>'IN BYTE. '
6312   032536    020056   033463   004467
6313   032544    047111   041040   052131
6314   032552    027105   000040
6315
6316   032556    051200   040505   020104  MTB45:  .ASCIZ  <CRLF>'READ DATA. '
6317   032564    040504   040524   020056
6318   032572       000
6319
6320   032573       011   047111   053440  MTC45:  .ASCIZ  <TAB>'IN WORD. '
6321   032600    051117   027104   000040
6322
6323   032606    053600   047522   042524  MTA50:  .ASCIZ  <CRLF>'WROTE. 000'<TAB>'IN BYTE. '
6324   032614    020056   030060   004460
6325   032622    047111   041040   052131
6326   032630    027105   000040
6327
6328   032634    042600   052116   051105  PDMSG1: .ASCII  <CRLF>'ENTERING CACHE ADDRESS MEMORY POWER UP '
6329   032642    047111   020107   040503
6330   032650    044103   020105   042101
6331   032656    051104   051505   020123
6332   032664    042515   047515   054522
6333   032672    050040   053517   051105
6334   032700    052440   020120
6335   032704    047111   040526   044514          .ASCII  'INVALIDATOR TEST.'<CRLF>
6336   032712    040504   047524   020122
6337   032720    042524   052123   100056
6338   032726    046120   040505   042523          .ASCII  'PLEASE GO THROUGH A POWER DOWN, POWER UP '
6339   032734    043440   020117   044124
6340   032742    047522   043525   020110
6341   032750    020101   047520   042527
6342   032756    020122   047504   047127
6343   032764    020054   047520   042527
6344   032772    020122   050125       040
6345   032777       123   050505   042525          .ASCIZ  'SEQUENCE.'<CRLF>
6346   033004    041516   027105   000200
6347
6348   033012    041600   041501   042510  PDMSG2: .ASCII  <CRLF>'CACHE ADDRESS MEMORY POWER UP INVALIDATOR'
6349   033020    040440   042104   042522
6350   033026    051523   046440   046505
6351   033034    051117   020131   047520
6352   033042    042527   020122   050125
6353   033050    044440   053116   046101
6354   033056    042111   052101   051117
```

```
6355  033064  052040  051505  020124          .ASCIZ  ' TEST DID NOT FAIL.'<CRLF>
6356  033072  044504  020104  047516
6357  033100  020124  040506  046111
6358  033106  100056     000
6359
6360  033111     105  051122  051117  ADRNG:  .ASCII  'ERROR ADDRESS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6361  033116  040440  042104  042522
6362  033124  051523  051040  043505
6363  033132  051511  042524  020122
6364  033140  042516  042105  042105
6365  033146  043040  051117  052040
6366  033154  051505  026124  041200
6367  033162  052125  044440  020124
6368  033170  040510  020123  042502
6369  033176  047105     040
6370  033201     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
6371  033206  042105  040440  020123
6372  033214  040502  020504     000
6373
6374  033221     105  051122  051117  ERRNG:  .ASCII  'ERROR REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6375  033226  051040  043505  051511
6376  033234  042524  020122  042516
6377  033242  042105  042105  043040
6378  033250  051117  052040  051505
6379  033256  026124  041200  052125
6380  033264  044440  020124  040510
6381  033272  020123  042502  047105
6382  033300     040
6383  033301     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
6384  033306  042105  040440  020123
6385  033314  040502  020504     000
6386
6387  033321     103  047117  051124  CNRNG:  .ASCII  'CONTROL REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6388  033326  046117  051040  043505
6389  033334  051511  042524  020122
6390  033342  042516  042105  042105
6391  033350  043040  051117  052040
6392  033356  051505  026124  041200
6393  033364  052125  044440  020124
6394  033372  040510  020123  042502
6395  033400  047105     040
6396  033403     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
6397  033410  042105  040440  020123
6398  033416  040502  020504     000
6399  033423     115  044501  052116  MNRNG:  .ASCII  'MAINTENANCE REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
6400  033430  047105  047101  042503
6401  033436  051040  043505  051511
6402  033444  042524  020122  042516
6403  033452  042105  042105  043040
6404  033460  051117  052040  051505
6405  033466  026124  041200  052125
6406  033474  044440  020124  040510
6407  033502  020123  042502  047105
6408  033510     040
6409  033511     106  040514  043507          .ASCIZ  'FLAGGED AS BAD!'
6410  033516  042105  040440  020123
```

```
  6411  033524  040502  020504     000
  6412
  6413  033531     110  052111  046457  MMRNG:  .ASCII  'HIT/MISS REGISTER NEEDED FOR TEST,'<CRLF>'BUT IT HAS BEEN '
  6414  033536  051511  020123  042522
  6415  033544  044507  052123  051105
  6416  033552  047040  042505  042504
  6417  033560  020104  047506  020122
  6418  033566  042524  052123  100054
  6419  033574  052502  020124  052111
  6420  033602  044040  051501  041040
  6421  033610  042505  020116
  6422  033614  046106  043501  042507          .ASCIZ  'FLAGGED AS BAD!'
  6423  033622  020104  051501  041040
  6424  033630  042101  000041
  6425
  6426  033634  040600  042104  042522  MTA77:  .ASCIZ  <CRLF>'ADDRESS:  '
  6427  033642  051523  020072  000040
  6428
  6429  033650  051440  047510  046125  MTB77:  .ASCIZ  ' SHOULD HAVE BEEN A HIT IN GROUP '
  6430  033656  020104  040510  042526
  6431  033664  041040  042505  020116
  6432  033672  020101  044510  020124
  6433  033700  047111  043440  047522
  6434  033706  050125  000040
  6435
  6436  033712  043101  042524  020122  MTC77:  .ASCIZ  'AFTER REFERENCING'<CRLF>'ADDRESS:  '
  6437  033720  042522  042506  042522
  6438  033726  041516  047111  100107
  6439  033734  042101  051104  051505
  6440  033742  035123  020040     000
  6441
  6442  033747     040  044127  046111  MTD77:  .ASCIZ  ' WHILE FORCING SELECTION OF GROUP '
  6443  033754  020105  047506  041522
  6444  033762  047111  020107  042523
  6445  033770  042514  052103  047511
  6446  033776  020116  043117  043440
  6447  034004  047522  050125  000040
  6448
  6449  034012  040600  051122  051117  MTA101: .ASCII  <CRLF>'ARROR ADRS REG.'<TAB>'ERROR REG.'<TAB>
  6450  034020  040440  051104  020123
  6451  034026  042522  027107  042411
  6452  034034  051122  051117  051040
  6453  034042  043505  004456
  6454  034046  054105  042520  052103          .ASCIZ  'EXPECTED ERR.'<TAB>'PATTERN PUT IN MAINT REG.'<CRLF>
  6455  034054  042105  042440  051122
  6456  034062  004456  040520  052124
  6457  034070  051105  020116  052520
  6458  034076  020124  047111  046440
  6459  034104  044501  052116  051040
  6460  034112  043505  100056     000
  6461
  6462  034117     200  043101  042524  MTA120: .ASCIZ  <CRLF>'AFTER 2ND CYCLE READ  '
  6463  034124  020122  047062  020104
  6464  034132  054503  046103  020105
  6465  034140  042522  042101  020040
  6466  034146     000
```

```
6467
6468  034147     200  043101  042524  MTB120: .ASCIZ  <CRLF>'AFTER 4TH CYCLE READ  '
6469  034154  020122  052064  020110
6470  034162  054503  046103  020105
6471  034170  042522  042101  020040
6472  034176     000
6473
6474  034177     200  043101  042524  MTC120: .ASCIZ  <CRLF>'AFTER 6TH CYCLE READ  '
6475  034204  020122  052066  020110
6476  034212  054503  046103  020105
6477  034220  042522  042101  020040
6478  034226     000
6479  034227     200  043101  042524  MTD120: .ASCIZ  <CRLF>'AFTER 8TH CYCLE READ  '
6480  034234  020122  052070  020110
6481  034242  054503  046103  020105
6482  034250  042522  042101  020040
6483  034256     000
6484
6485  034257     200  043101  042524  MTE120: .ASCIZ  <CRLF>'AFTER 10TH CYCLE READ '
6486  034264  020122  030061  044124
6487  034272  041440  041531  042514
6488  034300  051040  040505  020104
6489  034306     000
6490
6491  034307     200  043101  042524  MTF120: .ASCIZ  <CRLF>'AFTER 12TH CYCLE READ '
6492  034314  020122  031061  044124
6493  034322  041440  041531  042514
6494  034330  051040  040505  020104
6495  034336     000
6496
6497  034337     106  047522  020115  MTG120: .ASCIZ  'FROM THE HIT/MISS REG. EXPECTED '
6498  034344  044124  020105  044510
6499  034352  027524  044515  051523
6500  034360  051040  043505  020056
6501  034366  054105  042520  052103
6502  034374  042105  000040
6503
6504  034400  052200  042510  050040  MTA124: .ASCII  <CRLF>'THE PATTERN BEING USED IN THE MAINTENANCE '
6505  034406  052101  042524  047122
6506  034414  041040  044505  043516
6507  034422  052440  042523  020104
6508  034430  047111  052040  042510
6509  034436  046440  044501  052116
6510  034444  047105  047101  042503
6511  034452     040
6512  034453     122  043505  051511          .ASCIZ  'REGISTER WAS: '
6513  034460  042524  020122  040527
6514  034466  035123  000040
6515
6516  034472  051200  043105  051105  MTA126: .ASCIZ  <CRLF>'REFERENCED ADDRESS:'<TAB>
6517  034500  047105  042503  020104
6518  034506  042101  051104  051505
6519  034514  035123  000011
6520
6521  034520  040600  051122  051117  MTB126: .ASCIZ  <CRLF>'ARROR ADDRESS REGISTER:'<TAB>
6522  034526  040440  042104  042522
```

```
6523  034534  051523  051040  043505
6524  034542  051511  042524  035122
6525  034550  000011
6526
6527  034552  050200  052101  042524  MTA131: .ASCIZ  <CRLF>'PATTERN BEING USED IN THE MAINTENANCE REGISTER:'<TAB>
6528  034560  047122  041040  044505
6529  034566  043516  052440  042523
6530  034574  020104  047111  052040
6531  034602  042510  046440  044501
6532  034610  052116  047105  047101
6533  034616  042503  051040  043505
6534  034624  051511  042524  035122
6535  034632  000011
6536
6537  034634  042600  050130  041505  MTB131: .ASCIZ  <CRLF>'EXPECTED ERROR REGISTER:'<TAB>
6538  034642  042524  020104  051105
6539  034650  047522  020122  042522
6540  034656  044507  052123  051105
6541  034664  004472     000
6542
6543  034667     200  047507  020124  MTC131: .ASCIZ  <CRLF>'GOT ERROR REGISTER:'<TAB>
6544  034674  051105  047522  020122
6545  034702  042522  044507  052123
6546  034710  051105  004472     000
6547
6548  034715     200  051105  047522  MTA134: .ASCIZ  <CRLF>'ERROR ADR REG.'<TAB>'ERROR REG.'<CRLF>
6549  034722  020122  042101  020122
6550  034730  042522  027107  042411
6551  034736  051122  051117  051040
6552  034744  043505  100056     000
6553
6554  034751     200  054105  042520  MTA135: .ASCIZ  <CRLF>'EXPECTED ERROR REG.:  '
6555  034756  052103  042105  042440
6556  034764  051122  051117  051040
6557  034772  043505  035056  020040
6558  035000     000
6559
6560  035001     107  052117  042440  MTB135: .ASCIZ  'GOT ERROR REG.:  '
6561  035006  051122  051117  051040
6562  035014  043505  035056  020040
6563  035022     000
6564
6565  035023     200  054105  042520  MTC135: .ASCIZ  <CRLF>'EXPECTED ERROR ADR REG.:  '
6566  035030  052103  042105  042440
6567  035036  051122  051117  040440
6568  035044  051104  051040  043505
6569  035052  035056  020040     000
6570
6571  035057     107  052117  042440  MTD135: .ASCIZ  'GOT ERROR ADR REG.:  '
6572  035064  051122  051117  040440
6573  035072  051104  051040  043505
6574  035100  035056  020040     000
6575
6576
6577                                  ;THESE ARE THE ERROR MESSAGES:
6578
```

```
6579  035105    101  051040  043105  EM1:    .ASCIZ  'A REFERENCE WHICH SHOULD HAVE BEEN A HIT WAS A MISS.'
6580  035112  051105  047105  042503
6581  035120  053440  044510  044103
6582  035126  051440  047510  046125
6583  035134  020104  040510  042526
6584  035142  041040  042505  020116
6585  035150  020101  044510  020124
6586  035156  040527  020123  020101
6587  035164  044515  051523  000056
6588
6589
6590  035172  052600  042516  050130  EM14:   .ASCIZ  <CRLF>'UNEXPECTED PARITY ERROR TRAP.'
6591  035200  041505  042524  020104
6592  035206  040520  044522  054524
6593  035214  042440  051122  051117
6594  035222  052040  040522  027120
6595  035230     000
6596
6597  035231     052  025052  042524  EM15:   .ASCIZ  '***TEST ABORTED! GOING TO NEXT TEST.***'
6598  035236  052123  04044C  047502
6599  035244  052122  042105  020041
6600  035252  047507  047111  020107
6601  035260  047524  047040  054105
6602  035266  020124  042524  052123
6603  035274  025056  025052     000
6604  035301     103  041501  042510  EM55:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
6605  035306  051040  043505  051511
6606  035314  042524  020122  042522
6607  035322  050123  047117  042523
6608  035330  052040  051505  020124
6609  035336  040506  046111  042105
6610  035344  100056
6611  035346  020101  042522  042506          .ASCII  'A REFERENCE TO THE LOW ORDER ERROR ADDRESS REGISTER '
6612  035354  042522  041516  020105
6613  035362  047524  052040  042510
6614  035370  046040  053517  047440
6615  035376  042122  051105  042440
6616  035404  051122  051117  040440
6617  035412  042104  042522  051523
6618  035420  051040  043505  051511
6619  035426  042524  020122
6620  035432  044524  042515  020104          .ASCIZ  'TIMED OUT.'
6621  035440  052517  027124     000
6622
6623  035445     103  041501  042510  EM56:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
6624  035452  051040  043505  051511
6625  035460  042524  020122  042522
6626  035466  050123  047117  042523
6627  035474  052040  051505  020124
6628  035502  040506  046111  042105
6629  035510  100056
6630  035512  020101  042522  042506          .ASCII  'A REFERENCE TO THE HIGH ORDER ERROR ADDRESS REGISTER '
6631  035520  042522  041516  020105
6632  035526  047524  052040  042510
6633  035534  044040  043511  020110
6634  035542  051117  042504  020122
```

```
 6635  035550  051105  047522  020122
 6636  035556  042101  051104  051505
 6637  035564  020123  042522  044507
 6638  035572  052123  051105     040
 6639  035577     124  046511  042105          .ASCIZ  'TIMED OUT.'
 6640  035604  047440  052125  000056
 6641
 6642  035612  040503  044103  020105  EM57:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6643  035620  042522  044507  052123
 6644  035626  051105  051040  051505
 6645  035634  047520  051516  020105
 6646  035642  042524  052123  043040
 6647  035650  044501  042514  027104
 6648  035656     200
 6649  035657     101  051040  043105          .ASCIZ  'A REFERENCE TO THE ERROR REGISTER TIMED OUT.'
 6650  035664  051105  047105  042503
 6651  035672  052040  020117  044124
 6652  035700  020105  051105  047522
 6653  035706  020122  042522  044507
 6654  035714  052123  051105  052040
 6655  035722  046511  042105  047440
 6656  035730  052125  000056
 6657
 6658  035734  040503  044103  020105  EM60:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6659  035742  042522  044507  052123
 6660  035750  051105  051040  051505
 6661  035756  047520  051516  020105
 6662  035764  042524  052123  043040
 6663  035772  044501  042514  027104
 6664  036000     200
 6665  036001     101  051040  043105          .ASCIZ  'A REFERENCE TO THE CONTROL REGISTER TIMED OUT.'
 6666  036006  051105  047105  042503
 6667  036014  052040  020117  044124
 6668  036022  020105  047503  052116
 6669  036030  047522  020114  042522
 6670  036036  044507  052123  051105
 6671  036044  052040  046511  042105
 6672  036052  047440  052125  000056
 6673
 6674  036060  040503  044103  020105  EM61:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6675  036066  042522  044507  052123
 6676  036074  051105  051040  051505
 6677  036102  047520  051516  020105
 6678  036110  042524  052123  043040
 6679  036116  044501  042514  027104
 6680  036124     200
 6681  036125     101  051040  043105          .ASCIZ  'A REFERENCE TO THE MAINTENANCE REGISTER TIMED OUT.'
 6682  036132  051105  047105  042503
 6683  036140  052040  020117  044124
 6684  036146  020105  040515  047111
 6685  036154  042524  040516  041516
 6686  036162  020105  042522  044507
 6687  036170  052123  051105  052040
 6688  036176  046511  042105  047440
 6689  036204  052125  000056
 6690
```

```
 6691  036210  040503  044103  020105  EM62:   .ASCII  'CACHE REGISTER RESPONSE TEST FAILED.'<CRLF>
 6692  036216  042522  044507  052123
 6693  036224  051105  051040  051505
 6694  036232  047520  051516  020105
 6695  036240  042524  052123  043040
 6696  036246  044501  042514  027104
 6697  036254     200
 6698  036255     101  051040  043105          .ASCIZ  'A REFERENCE TO THE HIT/MISS REGISTER TIMED OUT.'<CRLF>
 6699  036262  051105  047105  042503
 6700  036270  052040  020117  044124
 6701  036276  020105  044510  027524
 6702  036304  044515  051523  051040
 6703  036312  043505  051511  042524
 6704  036320  020122  044524  042515
 6705  036326  020104  052517  027124
 6706  036334  000200
 6707
 6708  036336  040503  044103  020105  EM63:   .ASCII  'CACHE REGISTER DATA PATHS, READ ZEROES, TEST FAILED.'
 6709  036344  042522  044507  052123
 6710  036352  051105  042040  052101
 6711  036360  020101  040520  044124
 6712  036366  026123  051040  040505
 6713  036374  020104  042532  047522
 6714  036402  051505  020054  042524
 6715  036410  052123  043040  044501
 6716  036416  042514  027104
 6717  036422  053600  047522  042524          .ASCII  <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA '
 6718  036430  055040  051105  042517
 6719  036436  020123  052502  020124
 6720  036444  042522  042101  041040
 6721  036452  041501  020113  047516
 6722  036460  026516  042532  047522
 6723  036466  042040  052101  020101
 6724  036474  051106  046517  041040                .ASCIZ  'FROM BOTH'<CRLF>'THE CONTROL AND MAINTENANCE REGISTERS.'
 6725  036502  052117  100110  044124
 6726  036510  020105  047503  052116
 6727  036516  047522  020114  047101
 6728  036524  020104  040515  047111
 6729  036532  042524  040516  041516
 6730  036540  020105  042522  044507
 6731  036546  052123  051105  027123
 6732  036554     000
 6733
 6734  036555     103  041501  042510  EM64:   .ASCII  'CACHE REGISTER DATA PATH, READ ZEROES, TEST FAILED.'
 6735  036562  051040  043505  051511
 6736  036570  042524  020122  040504
 6737  036576  040524  050040  052101
 6738  036604  026110  051040  040505
 6739  036612  020104  042532  047522
 6740  036620  051505  020054  042524
 6741  036626  052123  043040  044501
 6742  036634  042514  027104
 6743  036640  053600  047522  042524          .ASCII  <CRLF>'WROTE ZEROES BUT READ BACK NON-ZERO DATA FROM '
 6744  036646  055040  051105  042517
 6745  036654  020123  052502  020124
 6746  036662  042522  042101  041040
```

```
 6747  036670  041501  020113  047516
 6748  036676  026516  042532  047522
 6749  036704  042040  052101  020101
 6750  036712  051106  046517     040
 6751  036717     200  044124  020105          .ASCIZ  <CRLF>'THE MAINTENANCE REGISTER.'
 6752  036724  040515  047111  042524
 6753  036732  040516  041516  020105
 6754  036740  042522  044507  052123
 6755  036746  051105  000056
 6756
 6757  036752  040503  044103  020105  EM65:   .ASCII  'CACHE REGISTER DATA PATHS, READ ONES, REST FAILED.'<CRLF>
 6758  036760  042522  044507  052123
 6759  036766  051105  042040  052101
 6760  036774  020101  040520  044124
 6761  037002  026123  051040  040505
 6762  037010  020104  047117  051505
 6763  037016  020054  042522  052123
 6764  037024  043040  044501  042514
 6765  037032  027104     200
 6766  037035     106  044501  042514          .ASCII  'FAILED TO READ CORRECT DATA FROM THE ADDRESS REGISTER'
 6767  037042  020104  047524  051040
 6768  037050  040505  020104  047503
 6769  037056  051122  041505  020124
 6770  037064  040504  040524  043040
 6771  037072  047522  020115  044124
 6772  037100  020105  042101  051104
 6773  037106  051505  020123  042522
 6774  037114  044507  052123  051105
 6775  037122  044440  020116  044124          .ASCII  ' IN THE CLEAR STATE.'<CRLF>'THE LOW ORDER ADDRESS '
 6776  037130  020105  046103  040505
 6777  037136  020122  052123  052101
 6778  037144  027105  052200  042510
 6779  037152  046040  053517  047440
 6780  037160  042122  051105  040440
 6781  037166  042104  042522  051523
 6782  037174     040
 6783  037175     123  047510  046125          .ASCII  'SHOULD HAVE BEEN SET TO: 177740'<CRLF>
 6784  037202  020104  040510  042526
 6785  037210  041040  042505  020116
 6786  037216  042523  020124  047524
 6787  037224  020072  033461  033467
 6788  037232  030064     200
 6789  037235     124  042510  044040          .ASCII  'THE HIGH ORDER ADDRESS REGISTER SHOULD HAVE BEEN '
 6790  037242  043511  020110  051117
 6791  037250  042504  020122  042101
 6792  037256  051104  051505  020123
 6793  037264  042522  044507  052123
 6794  037272  051105  051440  047510
 6795  037300  046125  020104  040510
 6796  037306  042526  041040  042505
 6797  037314  020116
 6798  037316  042523  020124  047524          .ASCIZ  'SET TO: 000003'
 6799  037324  020072  030060  030060
 6800  037332  031460     000
 6801
 6802  037335     103  041501  042510  EM66:   .ASCIZ  'CACHE CONTROL REGISTER COUNT PATTERN TEST FAILED.'
```

```
 6803   037342   041440   047117   051124
 6804   037350   046117   051040   043505
 6805   037356   051511   042524   020122
 6806   037364   047503   047125   020124
 6807   037372   040520   052124   051105
 6808   037400   020116   042524   052123
 6809   037406   043040   044501   042514
 6810   037414   027104      000
 6811
 6812   037417      103   041501   042510  EM67:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 6813   037424   044040   052111   046457
 6814   037432   051511   020123   047101
 6815   037440   020104   047503   052116
 6816   037446   047522   020114   042522
 6817   037454   044507   052123   051105
 6818   037462   052040   051505   020124
 6819   037470   040506   046111   042105
 6820   037476      056
 6821   037477      200   044527   044124          .ASCII  <CRLF>'WITH THE CONTROL REGISTER CLEAR, THE HIT/MISS '
 6822   037504   052040   042510   041440
 6823   037512   047117   051124   046117
 6824   037520   051040   043505   051511
 6825   037526   042524   020122   046103
 6826   037534   040505   026122   052040
 6827   037542   042510   044040   052111
 6828   037550   046457   051511   020123
 6829   037556   042522   044507   052123          .ASCIZ  'REGISTER SHOULD'<CRLF>'HAVE SHOWN SIX HITS (000077).'
 6830   037564   051105   051440   047510
 6831   037572   046125   100104   040510
 6832   037600   042526   051440   047510
 6833   037606   047127   051440   054111
 6834   037614   044040   052111   020123
 6835   037622   030050   030060   033460
 6836   037630   024467   000056
 6837
 6838   037634   040503   044103   020105  EM70:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 6839   037642   044510   027524   044515
 6840   037650   051523   040440   042116
 6841   037656   041440   047117   051124
 6842   037664   046117   051040   043505
 6843   037672   051511   042524   020122
 6844   037700   042524   052123   043040
 6845   037706   044501   042514   027104
 6846   037714   053600   044510   042514          .ASCII  <CRLF>'WHILE FORCING SELECTION OF GROUP 1 AND FORCING '
 6847   037722   043040   051117   044503
 6848   037730   043516   051440   046105
 6849   037736   041505   044524   047117
 6850   037744   047440   020106   051107
 6851   037752   052517   020120   020061
 6852   037760   047101   020104   047506
 6853   037766   041522   047111   020107
 6854   037774   044515   051523   051505          .ASCII  'MISSES TO GROUP 0,'<CRLF>'THE HIT/MISS REGISTER '
 6855   040002   052040   020117   051107
 6856   040010   052517   020120   026060
 6857   040016   052200   042510   044040
 6858   040024   052111   046457   051511
```

```
 6859  040032  020123  042522  044507
 6860  040040  052123  051105     040
 6861  040045     123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX HITS (000077).'
 6862  040052  020104  040510  042526
 6863  040060  051440  047510  047127
 6864  040066  051440  054111  044040
 6865  040074  052111  020123  030050
 6866  040102  030060  033460  024467
 6867  040110  000056
 6868
 6869  040112  040503  044103  020105  EM71:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 6870  040120  044510  027524  044515
 6871  040126  051523  040440  042116
 6872  040134  041440  047117  051124
 6873  040142  046117  051040  043505
 6874  040150  051511  042524  020122
 6875  040156  042524  052123  043040
 6876  040164  044501  042514  027104
 6877  040172  053600  044510  042514          .ASCII  <CRLF>'WHILE FORCING SELECTION OF GROUP 0 AND FORCING '
 6878  040200  043040  051117  044503
 6879  040206  043516  051440  046105
 6880  040214  041505  044524  047117
 6881  040222  047440  020106  051107
 6882  040230  052517  020120  020060
 6883  040236  047101  020104  047506
 6884  040244  041522  047111  020107
 6885  040252  044515  051523  051505          .ASCII  'MISSES TO GROUP 1,'<CRLF>'THE HIT/MISS REGISTER '
 6886  040260  052040  020117  051107
 6887  040266  052517  020120  026061
 6888  040274  052200  042510  044040
 6889  040302  052111  046457  051511
 6890  040310  020123  042522  044507
 6891  040316  052123  051105     040
 6892  040323     123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX HITS (000077).'
 6893  040330  020104  040510  042526
 6894  040336  051440  047510  047127
 6895  040344  051440  054111  044040
 6896  040352  052111  020123  030050
 6897  040360  030060  033460  024467
 6898  040366  000056
 6899
 6900  040370  040503  044103  020105  EM72:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 6901  040376  044510  027524  044515
 6902  040404  051523  040440  042116
 6903  040412  041440  047117  051124
 6904  040420  046117  051040  043505
 6905  040426  051511  042524  020122
 6906  040434  042524  052123  043040
 6907  040442  044501  042514  027104
 6908  040450  044127  046111  020105          .ASCII  'WHILE FORCING MISSES TO BOTH GROUPS, THE HIT/MISS '
 6909  040456  047506  041522  047111
 6910  040464  020107  044515  051523
 6911  040472  051505  052040  020117
 6912  040500  047502  044124  043440
 6913  040506  047522  050125  026123
 6914  040514  052040  042510  044040
```

```
 6915  040522  052111  046457  051511
 6916  040530  020123
 6917  040532  042522  044507  052123          .ASCIZ  'REGISTER'<CRLF>'SHOULD HAVE SHOWN SIX MISSES (000000).'
 6918  040540  051105  051600  047510
 6919  040546  046125  020104  040510
 6920  040554  042526  051440  047510
 6921  040562  047127  051440  054111
 6922  040570  046440  051511  042523
 6923  040576  020123  030050  030060
 6924  040604  030060  024460  000056
 6925
 6926  040612  040503  044103  020105  EM73:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 6927  040620  044510  027524  044515
 6928  040626  051523  040440  042116
 6929  040634  041440  047117  051124
 6930  040642  046117  051040  043505
 6931  040650  051511  042524  020122
 6932  040656  042524  052123  043040
 6933  040664  044501  042514  027104
 6934  040672  053600  04451C  042514          .ASCII  <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '
 6935  040700  043040  051117  044503
 6936  040706  043516  046440  051511
 6937  040714  042523  020123  047524
 6938  040722  041040  052117  020110
 6939  040730  051107  052517  051520
 6940  040736  040440  042116  043040
 6941  040744  051117  044503  043516
 6942  040752     040
 6943  040753     123  046105  041505          .ASCII  'SELECTION OF GROUP 1,'<CRLF>'THE HIT/MISS REGISTER '
 6944  040760  044524  047117  047440
 6945  040766  020106  051107  052517
 6946  040774  020120  026061  052200
 6947  041002  042510  044040  052111
 6948  041010  046457  051511  020123
 6949  041016  042522  044507  052123
 6950  041024  051105     040
 6951  041027     123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX MISSES (000000).'
 6952  041034  020104  040510  042526
 6953  041042  051440  047510  047127
 6954  041050  051440  054111  046440
 6955  041056  051511  042523  020123
 6956  041064  030050  030060  030060
 6957  041072  024460  000056
 6958
 6959  041076  040503  044103  020105  EM74:   .ASCII  'CACHE HIT/MISS AND CONTROL REGISTER TEST FAILED.'
 6960  041104  044510  027524  044515
 6961  041112  051523  040440  042116
 6962  041120  041440  047117  051124
 6963  041126  046117  051040  043505
 6964  041134  051511  042524  020122
 6965  041142  042524  052123  043040
 6966  041150  044501  042514  027104
 6967  041156  053600  044510  042514          .ASCII  <CRLF>'WHILE FORCING MISSES TO BOTH GROUPS AND FORCING '
 6968  041164  043040  051117  044503
 6969  041172  043516  046440  051511
 6970  041200  042523  020123  047524
```

```
6971   041206  041040  052117  020110
6972   041214  051107  052517  051520
6973   041222  040440  042116  043040
6974   041230  051117  044503  043516
6975   041236     040
6976   041237     123  046105  041505          .ASCII  'SELECTION OF GROUP 0.'<CRLF>'THE HIT/MISS REGISTER '
6977   041244  044524  047117  047440
6978   041252  020106  051107  052517
6979   041260  020120  026060  052200
6980   041266  042510  044040  052111
6981   041274  046457  051511  020123
6982   041302  042522  044507  052123
6983   041310  051105     040
6984   041313     123  047510  046125          .ASCIZ  'SHOULD HAVE SHOWN SIX MISSES (000000).'
6985   041320  020104  040510  042526
6986   041326  051440  047510  047127
6987   041334  051440  054111  046440
6988   041342  051511  042523  020123
6989   041350  030050  030060  030060
6990   041356  024460  000056
6991
6992   041362  047503  052116  047522  EM75:   .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>'FAILED TO GET '
6993   041370  020114  042522  044507
6994   041376  052123  051105  052040
6995   041404  051505  020124  040506
6996   041412  046111  042105  100056
6997   041420  040506  046111  042105
6998   041426  052040  020117  042507
6999   041434  020124
7000   041436  020101  044510  020124          .ASCIZ  'A HIT ON A REFERENCE WHICH SHOULD HAVE BEEN A HIT.'
7001   041444  047117  040440  051040
7002   041452  043105  051105  047105
7003   041460  042503  053440  044510
7004   041466  044103  051440  047510
7005   041474  046125  020104  040510
7006   041502  042526  041040  042505
7007   041510  020116  020101  044510
7008   041516  027124     000
7009
7010           041362                  EM76=EM75
7011
7012   041521     103  047117  051124  EM77:   .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>'THE WRONG '
7013   041526  046117  051040  043505
7014   041534  051511  042524  020122
7015   041542  042524  052123  043040
7016   041550  044501  042514  027104
7017   041556  052200  042510  053440
7018   041564  047522  043516     040
7019   041571     107  047522  050125          .ASCIZ  'GROUP WAS WRITTEN WHILE FORCING SELECTION OF A GROUP.'
7020   041576  053440  051501  053440
7021   041604  044522  052124  047105
7022   041612  053440  044510  042514
7023   041620  043040  051117  044503
7024   041626  043516  051440  046105
7025   041634  041505  044524  047117
7026   041642  047440  020106  020101
```

```
 7027  041650  051107  052517  027120
 7028  041656     000
 7029
 7030  041657     103  047117  051124  EM117:  .ASCII  'CONTROL REGISTER TEST FAILED.'<CRLF>
 7031  041664  046117  051040  043505
 7032  041672  051511  042524  020122
 7033  041700  042524  052123  043040
 7034  041706  044501  042514  027104
 7035  041714     200
 7036  041715     107  052117  040440          .ASCIZ  'GOT A HIT IN THE GROUP TO WHICH MISSES ARE BEING FORCED.'
 7037  041722  044040  052111  044440
 7038  041730  020116  044124  020105
 7039  041736  051107  052517  020120
 7040  041744  047524  053440  044510
 7041  041752  044103  046440  051511
 7042  041760  042523  020123  051101
 7043  041766  020105  042502  047111
 7044  041774  020107  047506  041522
 7045  042002  042105  000056
 7046
 7047  042006  044510  027524  044515  EM120:  .ASCII  'HIT/MISS REGISTER PATTERNS TEST FAILED.'
 7048  042014  051523  051040  043505
 7049  042022  051511  042524  020122
 7050  042030  040520  052124  051105
 7051  042036  051516  052040  051505
 7052  042044  020124  040506  046111
 7053  042052  042105     056
 7054  042055     200  042522  042101          .ASCII  <CRLF>'READ WRONG DATA FROM THE HIT/MISS REGISTER'<CRLF>
 7055  042062  053440  047522  043516
 7056  042070  042040  052101  020101
 7057  042076  051106  046517  052040
 7058  042104  042510  044040  052111
 7059  042112  046457  051511  020123
 7060  042120  042522  044507  052123
 7061  042126  051105     200
 7062  042131     127  044510  042514          .ASCIZ  'WHILE FLOATING A PATTERN OF HITS AND MISSES THROUGH IT.'
 7063  042136  043040  047514  052101
 7064  042144  047111  020107  020101
 7065  042152  040520  052124  051105
 7066  042160  020116  043117  044040
 7067  042166  052111  020123  047101
 7068  042174  020104  044515  051523
 7069  042202  051505  052040  051110
 7070  042210  052517  044107  044440
 7071  042216  027124     000
 7072
 7073  042221     103  041501  042510  EM121:  .ASCII  /CACHE CONTROL SIGNAL, THE 'RANDOM' SIGNAL, TEST FAILED./
 7074  042226  041440  047117  051124
 7075  042234  046117  051440  043511
 7076  042242  040516  026114  052040
 7077  042250  042510  023440  040522
 7078  042256  042116  046517  020047
 7079  042264  044523  047107  046101
 7080  042272  020054  042524  052123
 7081  042300  043040  044501  042514
 7082  042306  027104
```

```
 7093  042310  043200  044501  042514          .ASCII  <CRLF>'FAILED TO GET BOTH HITS AT THE TWO TEST ADDRESSES '
 7094  042316  020104  047524  043440
 7095  042324  052105  041040  052117
 7096  042332  020110  044510  051524
 7097  042340  040440  020124  044124
 7098  042346  020105  053524  020117
 7099  042354  042524  052123  040440
 7100  042362  042104  042522  051523
 7101  042370  051505     040
 7102  042373     127  044510  044103          .ASCIZ  'WHICH WERE REFERENCED.'
 7103  042400  053440  051105  020105
 7104  042406  042522  042506  042522
 7105  042414  041516  042105  000056
 7106
 7107  042422  040515  047111  042524  EM122:  .ASCII  'MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'
 7108  042430  040516  041516  020105
 7109  042436  042522  044507  052123
 7110  042444  051105  041440  052517
 7111  042452  052116  050040  052101
 7112  042460  042524  047122  052040
 7113  042466  051505  020124  040506
 7114  042474  046111  042105     056
 7115  042501     200  044124  020105          .ASCII  <CRLF>'THE MAINTENANCE REGISTER WILL NOT CLEAR.'
 7116  042506  040515  047111  042524
 7117  042514  040516  041516  020105
 7118  042522  042522  044507  052123
 7119  042530  051105  053440  046111
 7120  042536  020114  047516  020124
 7121  042544  046103  040505  027122
 7122
 7123  042552  040503  044103  020105  EM123:  .ASCII  'CACHE MAINTENANCE REGISTER COUNT PATTERN TEST FAILED.'
 7124  042560  040515  047111  042524
 7125  042566  040516  041516  020105
 7126  042574  042522  044507  052123
 7127  042602  051105  041440  052517
 7128  042610  052116  050040  052101
 7129  042616  042524  047122  052040
 7130  042624  051505  020124  040506
 7131  042632  046111  042105     056
 7132  042637     200  043101  042524          .ASCII  <CRLF>'AFTER WRITING A PATTERN IN THIS REGISTER '
 7133  042644  020122  051127  052111
 7134  042652  047111  020107  020101
 7135  042660  040520  052124  051105
 7136  042666  020116  047111  052040
 7137  042674  044510  020123  042522
 7138  042702  044507  052123  051105
 7139  042710     040
 7140  042711     106  044501  042514          .ASCIZ  'FAILED TO READ THAT PATTERN BACK.'
 7141  042716  020104  047524  051040
 7142  042724  040505  020104  044124
 7143  042732  052101  050040  052101
 7144  042740  042524  047122  041040
 7145  042746  041501  027113     000
 7146
 7147  042753     101  020116  047125  EM124:  .ASCII  'AN UNEXPECTED ERROR OCCURRED WHILE RUNNING THE '
 7148  042760  054105  042520  052103
```

```
 7139  042766  042105  042440  051122
 7140  042774  051117  047440  041503
 7141  043002  051125  042522  020104
 7142  043010  044127  046111  020105
 7143  043016  052522  047116  047111
 7144  043024  020107  044124  020105
 7145  043032  040515  047111  042524          .ASCII  'MAINTENANCE REGISTER'<CRLF>'COUNT PATTERN '
 7146  043040  040516  041516  020105
 7147  043046  042522  044507  052123
 7148  043054  051105  041600  052517
 7149  043062  052116  050040  052101
 7150  043070  042524  047122     040
 7151  043075     124  051505  027124          .ASCIZ  'TEST. NOTE MISSES WERE BEING FORCED TO BOTH GROUPS.'
 7152  043102  047040  052117  020105
 7153  043110  044515  051523  051505
 7154  043116  053440  051105  020105
 7155  043124  042502  047111  020107
 7156  043132  047506  041522  042105
 7157  043140  052040  020117  047502
 7158  043146  044124  043440  047522
 7159  043154  050125  027123     000
 7160
 7161  043161     115  044501  052116  EM127:  .ASCII  'MAINTENANCE REGISTER TEST FAILED.'<CRLF>
 7162  043166  047105  047101  042503
 7163  043174  051040  043505  051511
 7164  043202  042524  020122  042524
 7165  043210  052123  043040  044501
 7166  043216  042514  027104     200
 7167  043223     116  020117  051124          .ASCII  'NO TRAP OR ABORT OCCURRED WHEN THE PATTERN WAS PUT '
 7168  043230  050101  047440  020122
 7169  043236  041101  051117  020124
 7170  043244  041517  052503  051122
 7171  043252  042105  053440  042510
 7172  043260  020116  044124  020105
 7173  043266  040520  052124  051105
 7174  043274  020116  040527  020123
 7175  043302  052520  020124
 7176  043306  047111  052040  042510          .ASCIZ  'IN THE MAINTENANCE REGISTER.'
 7177  043314  046440  044501  052116
 7178  043322  047105  047101  042503
 7179  043330  051040  043505  051511
 7180  043336  042524  027122     000
 7181
 7182  043343     105  051122  051117  EM130:  .ASCIZ  'ERROR REGISTER WILL NOT UNLOCK, OR CLEAR.'
 7183  043350  051040  043505  051511
 7184  043356  042524  020122  044527
 7185  043364  046114  047040  052117
 7186  043372  052440  046116  041517
 7187  043400  026113  047440  020122
 7188  043406  046103  040505  027122
 7189  043414     000
 7190
 7191  043415     105  051122  051117  EM131:  .ASCII  'ERROR REGISTER AND MAINTENANCE REGISTER TEST FAILED.'
 7192  043422  051040  043505  051511
 7193  043430  042524  020122  047101
 7194  043436  020104  040515  047111
```

```
7195  043444  042524  040516  041516
7196  043452  020105  042522  044507
7197  043460  052123  051105  052040
7198  043466  051505  020124  040506
7199  043474  046111  042105     056
7200  043501     200  051105  047522          .ASCII  <CRLF>'ERROR REGISTER IS INCORRECTLY SET'
7201  043506  020122  042522  044507
7202  043514  052123  051105  044440
7203  043522  020123  047111  047503
7204  043530  051122  041505  046124
7205  043536  020131  042523     124
7206  043543     200  047506  020122          .ASCIZ  <CRLF>'FOR THE ERROR THAT WAS FORCED USING THE MAINTENANCE REGISTER.'
7207  043550  044124  020105  051105
7208  043556  047522  020122  044124
7209  043564  052101  053440  051501
7210  043572  043040  051117  042503
7211  043600  020104  051525  047111
7212  043606  020107  044124  020105
7213  043614  040515  047111  042524
7214  043622  040516  041516  020105
7215  043630  042522  044507  052123
7216  043636  051105  000056
7217
7218  043642                          EM140:
7219  043642  040515  047111  046440          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
7220  043650  046505  051117  020131
7221  043656  040504  040524  050040
7222  043664  051101  052111  020131
7223  043672  044103  041505  042513
7224  043700  051522  052040  051505
7225  043706  020124  040506  046111
7226  043714  042105     056
7227  043717     200  047125  041101          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
7228  043724  042514  052040  020117
7229  043732  047506  041522  020105
7230  043740  020101  040520  044522
7231  043746  054524  042440  051122
7232  043754  051117  020054  051525
7233  043762  047111  020107
7234  043766  044124  020105  040515          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
7235  043774  047111  042524  040516
7236  044002  041516  020105  042522
7237  044010  044507  052123  051105
7238  044016  100054
7239  044020  052101  052040  042510          .ASCII  'AT THE MAIN MEMORY EVEN WORD, LOW BYTE, PARITY '
7240  044026  046440  044501  020116
7241  044034  042515  047515  054522
7242  044042  042440  042526  020116
7243  044050  047527  042122  020054
7244  044056  047514  020127  054502
7245  044064  042524  020054  040520
7246  044072  044522  054524     040
7247  044077     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
7248  044104  051105  100054  051040
7249  044112  040505  044504  043516
7250  044120  040440  042040  052101
```

```
 7251  044126  020101  040520  052124
 7252  044134  051105  020116  044127
 7253  044142  041511  020110
 7254  044146  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7255  044154  044040  053101  020105
 7256  044162  040503  051525  042105
 7257  044170  040440  020116  051105
 7258  044176  047522  027122     000
 7259
 7260  044203                          EM141:
 7261  044203     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 7262  044210  042515  047515  054522
 7263  044216  042040  052101  020101
 7264  044224  040520  044522  054524
 7265  044232  041440  042510  045503
 7266  044240  051105  020123  042524
 7267  044246  052123  043040  044501
 7268  044254  042514  027104
 7269  044260  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR. USING '
 7270  044266  020105  047524  043040
 7271  044274  051117  042503  040440
 7272  044302  050040  051101  052111
 7273  044310  020131  051105  047522
 7274  044316  026122  052440  044523
 7275  044324  043516     040
 7276  044327     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 7277  044334  044501  052116  047105
 7278  044342  047101  042503  051040
 7279  044350  043505  051511  042524
 7280  044356  026122     200
 7281  044361     101  020124  044124          .ASCII  'AT THE MAIN MEMORY ODD WORD, LOW BYTE, PARITY '
 7282  044366  020105  040515  047111
 7283  044374  046440  046505  051117
 7284  044402  020131  042117  020104
 7285  044410  047527  042122  020054
 7286  044416  047514  020127  054502
 7287  044424  042524  020054  040520
 7288  044432  044522  054524     040
 7289  044437     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 7290  044444  051105  100054  051040
 7291  044452  040505  044504  043516
 7292  044460  040440  042040  052101
 7293  044466  020101  040520  052124
 7294  044474  051105  020116  044127
 7295  044502  041511  020110
 7296  044506  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7297  044514  044040  053101  020105
 7298  044522  040503  051525  042105
 7299  044530  040440  020116  051105
 7300  044536  047522  027122     000
 7301
 7302  044543                          EM142:
 7303  044543     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 7304  044550  042515  047515  054522
 7305  044556  042040  052101  020101
 7306  044564  040520  044522  054524
```

```
 7307  044572  041440  042510  045503
 7308  044600  051105  020123  042524
 7309  044606  052123  043040  044501
 7310  044614  042514  027104
 7311  044620  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 7312  044626  020105  047524  043040
 7313  044634  051117  042503  040440
 7314  044642  050040  051101  052111
 7315  044650  020131  051105  047522
 7316  044656  026122  052440  044523
 7317  044664  043516     040
 7318  044667     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 7319  044674  044501  052116  047105
 7320  044702  047101  042503  051040
 7321  044710  043505  051511  042524
 7322  044716  026122     200
 7323  044721     101  020124  044124          .ASCII  'AT THE MAIN MEMORY EVEN WORD, HIGH BYTE, PARITY '
 7324  044726  020105  040515  047111
 7325  044734  046440  046505  051117
 7326  044742  020131  053105  047105
 7327  044750  053440  051117  026104
 7328  044756  044040  043511  020110
 7329  044764  054502  042524  020054
 7330  044772  040520  044522  054524
 7331  045000     040
 7332  045001     103  042510  045503          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 7333  045006  051105  100054  051040
 7334  045014  040505  044504  043516
 7335  045022  040440  042040  052101
 7336  045030  020101  040520  052124
 7337  045036  051105  020116  044127
 7338  045044  041511  020110
 7339  045050  044123  052517  042114          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7340  045056  044040  053101  020105
 7341  045064  040503  051525  042105
 7342  045072  040440  020116  051105
 7343  045100  047522  027122     000
 7344
 7345  045105                          EM143:
 7346  045105     115  044501  020116          .ASCII  'MAIN MEMORY DATA PARITY CHECKERS TEST FAILED.'
 7347  045112  042515  047515  054522
 7348  045120  042040  052101  020101
 7349  045126  040520  044522  054524
 7350  045134  041440  042510  045503
 7351  045142  051105  020123  042524
 7352  045150  052123  043040  044501
 7353  045156  042514  027104
 7354  045162  052600  040516  046102          .ASCII  <CRLF> 'UNABLE TO FORCE A PARITY ERROR, USING '
 7355  045170  020105  047524  043040
 7356  045176  051117  042503  040440
 7357  045204  050040  051101  052111
 7358  045212  020131  051105  047522
 7359  045220  026122  052440  044523
 7360  045226  043516     040
 7361  045231     124  042510  046440          .ASCII  'THE MAINTENANCE REGISTER,'<CRLF>
 7362  045236  044501  052116  047105
```

```
 7363  045244  047101  042503  051040
 7364  045252  043505  051511  042524
 7365  045260  026122     200
 7366  045263     101  020124  044124          .ASCII  'AT THE MAIN MEMORY ODD WORD, HIGH BYTE, PARITY '
 7367  045270  020105  040515  047111
 7368  045276  046440  046505  051117
 7369  045304  020131  042117  020104
 7370  045312  047527  042122  020054
 7371  045320  044510  044107  041040
 7372  045326  052131  026105  050040
 7373  045334  051101  052111  020131
 7374  045342  044103  041505  042513          .ASCII  'CHECKER,'<CRLF>' READING A DATA PATTERN WHICH '
 7375  045350  026122  020200  042522
 7376  045356  042101  047111  020107
 7377  045364  020101  040504  040524
 7378  045372  050040  052101  042524
 7379  045400  047122  053440  044510
 7380  045406  044103     040
 7381  045411     123  047510  046125          .ASCIZ  'SHOULD HAVE CAUSED AN ERROR.'
 7382  045416  020104  04051C  042526
 7383  045424  041440  052501  042523
 7384  045432  020104  047101  042440
 7385  045440  051122  051117  000056
 7386
 7387  045446  020040  042524  052123  DH140:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'DATA.'<TAB>'ADDRESS.'
 7388  045454  004456  040503  046114
 7389  045462  040440  020124  041520
 7390  045470  004456  040504  040524
 7391  045476  004456  042101  051104
 7392  045504  051505  027123     000
 7393
 7394          045446                  DH141=DH140
 7395
 7396          045446                  DH142=DH140
 7397
 7398          045446                  DH143=DH140
 7399
 7400  045511     004     003     000  DF140:  .BYTE   4,3,0,2
 7401  045514     002
 7402
 7403          045511                  DF141=DF140
 7404
 7405          045511                  DF142=DF140
 7406
 7407          045511                  DF143=DF140
 7408
 7409          045516                          .EVEN
 7410  045516  001232  001116  001236  DT140:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
 7411  045524  001240  000000
 7412
 7413          045516                  DT141=DT140
 7414
 7415          045516                  DT142=DT140
 7416
 7417          045516                  DT143=DT140
 7418
```

```
7419
7420  045530  051105  047522  020122  EM132:  .ASCII  'ERROR REGISTER TEST WAS UNABLE TO CAUSE A TIME OUT,'
7421  045536  042522  044507  052123
7422  045544  051105  052040  051505
7423  045552  020124  040527  020123
7424  045560  047125  041101  042514
7425  045566  052040  020117  040503
7426  045574  051525  020105  020101
7427  045602  044524  042515  047440
7428  045610  052125     054
7429  045613     200  052101  040440          .ASCIZ  <CRLF>'AT AN ADDRESS WHICH SHOULD HAVE TIMED OUT.'
7430  045620  020116  042101  051104
7431  045626  051505  020123  044127
7432  045634  041511  020110  044123
7433  045642  052517  042114  044040
7434  045650  053101  020105  044524
7435  045656  042515  020104  052517
7436  045664  027124     000
7437
7438  045667     105  051122  051117  EM133:  .ASCII  'ERROR REGISTER TEST FAILED.'
7439  045674  051040  043505  051511
7440  045702  042524  020122  042524
7441  045710  052123  043040  044501
7442  045716  042514  027104
7443  045722  040600  052106  051105          .ASCII  <CRLF>'AFTER CAUSING A TIME OUT THE ERROR REGISTER SHOULD '
7444  045730  041440  052501  044523
7445  045736  043516  040440  052040
7446  045744  046511  020105  052517
7447  045752  020124  044124  020105
7448  045760  051105  047522  020122
7449  045766  042522  044507  052123
7450  045774  051105  051440  047510
7451  046002  046125  020104
7452  046006  040510  042526  041040          .ASCIZ  'HAVE BEEN SET TO : 000000.'
7453  046014  042505  020116  042523
7454  046022  020124  047524  035040
7455  046030  030040  030060  030060
7456  046036  027060     000
7457
7458  046041     103  047117  051124  EM134:  .ASCII  'CONTROL REGISTER, DISABLE TRAPS, TEST FAILED.'
7459  046046  046117  051040  043505
7460  046054  051511  042524  026122
7461  046062  042040  051511  041101
7462  046070  042514  052040  040522
7463  046076  051520  020054  042524
7464  046104  052123  043040  044501
7465  046112  042514  027104
7466  046116  040600  052040  040522          .ASCIZ  <CRLF>'A TRAP OCCURRED WITH BIT 0 SET IN THE CONTROL REGISTER.'
7467  046124  020120  041517  052503
7468  046132  051122  042105  053440
7469  046140  052111  020110  044502
7470  046146  020124  020060  042523
7471  046154  020124  047111  052040
7472  046162  042510  041440  047117
7473  046170  051124  046117  051040
7474  046176  043505  051511  042524
```

```
 7475  046204  027122     000
 7476
 7477  046207     105  051122  051117  EM135:  .ASCII  'ERROR REGISTER, LOCK UP, TEST FAILED.'
 7478  046214  051040  043505  051511
 7479  046222  042524  026122  046040
 7480  046230  041517  020113  050125
 7481  046236  020054  042524  052123
 7482  046244  043040  044501  042514
 7483  046252  027104
 7484  046254  040600  052106  051105          .ASCII  <CRLF>'AFTER FORCING MULTIPLE ERRORS, TWO, THE ERROR '
 7485  046262  043040  051117  044503
 7486  046270  043516  046440  046125
 7487  046276  044524  046120  020105
 7488  046304  051105  047522  051522
 7489  046312  020054  053524  026117
 7490  046320  052040  042510  042440
 7491  046326  051122  051117     040
 7492  046333     122  043505  051511          .ASCIZ  'REGISTERS WAS INSORRECTLY SET.'
 7493  046340  042524  051522  053440
 7494  046346  051501  044440  051516
 7495  046354  051117  042522  052103
 7496  046362  054514  051440  052105
 7497  046370  000056
 7498
 7499  046372  052600  042516  050130  EM150:  .ASCIZ  <CRLF>'UNEXPECTED CPU ERROR TRAPPED TO VECTOR ERRVEC (4)!'
 7500  046400  041505  042524  020104
 7501  046406  050103  020125  051105
 7502  046414  047522  020122  051124
 7503  046422  050101  042520  020104
 7504  046430  047524  053040  041505
 7505  046436  047524  020122  051105
 7506  046444  053122  041505  024040
 7507  046452  024464  000041
 7508
 7509                                  ;THESE ARE DATA HEADERS:
 7510
 7511  046456  020040  042524  052123  DH1:    .ASCIZ  '  TEST.'<TAB>' GROUP.'<TAB>'PHYSICAL ADDR.'<TAB>'CALL AT PC.'
 7512  046464  004456  043440  047522
 7513  046472  050125  004456  044120
 7514  046500  051531  041511  046101
 7515  046506  040440  042104  027122
 7516  046514  041411  046101  020114
 7517  046522  052101  050040  027103
 7518  046530     000
 7519  046531     040  052040  051505  DH14:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADDR REG.'
 7520  046536  027124  041411  046101
 7521  046544  020114  052101  050040
 7522  046552  027103  042411  051122
 7523  046560  051117  040440  042104
 7524  046566  020122  042522  027107
 7525  046574  052011  040522  020120          .ASCII  <TAB>'TRAP AT PC.'<TAB>
 7526  046602  052101  050040  027103
 7527  046610     011
 7528  046611     105  051122  051117          .ASCIZ  'ERROR REG.'
 7529  046616  051040  043505  000056
 7530
```

```
 7531  046624  020040  042524  052123  DH15:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
 7532  046632  004456  040503  046114
 7533  046640  040440  020124  041520
 7534  046646  000056
 7535
 7536  046650  020040  042524  052123  DH55:   .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'REG ADDRESS.'
 7537  046656  004456  051124  050101
 7538  046664  040440  020124  041520
 7539  046672  004456  040503  046114
 7540  046700  040440  020124  041520
 7541  046706  004456  042522  020107
 7542  046714  042101  051104  051505
 7543  046722  027123     000
 7544
 7545          046650                  DH56=DH55
 7546
 7547          046650                  DH57=DH55
 7548
 7549          046650                  DH60=DH55
 7550
 7551          046650                  DH61=DH55
 7552
 7553          046650                  DH62=DH55
 7554
 7555  046725     040  052040  051505  DH63:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'CONTROL.'
 7556  046732  027124  041411  046101
 7557  046740  020114  052101  050040
 7558  046746  027103  041411  047117
 7559  046754  051124  046117     056
 7560  046761     115  044501  052116          .ASCIZ  'MAINT.'<TAB>'(DATA READ FROM EACH REGISTER)'
 7561  046766  004456  042050  052101
 7562  046774  020101  042522  042101
 7563  047002  043040  047522  020115
 7564  047010  040505  044103  051040
 7565  047016  043505  051511  042524
 7566  047024  024522     000
 7567
 7568  047027     040  052040  051505  DH64:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'CONTROL REGISTER DATA.'
 7569  047034  027124  041411  046101
 7570  047042  020114  052101  050040
 7571  047050  027103  041411  047117
 7572  047056  051124  046117  051040
 7573  047064  043505  051511  042524
 7574  047072  020122  040504  040524
 7575  047100  000056
 7576
 7577  047102  020040  042524  052123  DH65:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'LOW ORD.'<TAB>'HIGH ORD.'
 7578  047110  004456  040503  046114
 7579  047116  040440  020124  041520
 7580  047124  004456  047514  020127
 7581  047132  051117  027104  044011
 7582  047140  043511  020110  051117
 7583  047146  027104
 7584  047150  024011  040504  040524          .ASCIZ  <TAB>'(DATA READ FROM ADR. REG.)'
 7585  047156  051040  040505  020104
 7586  047164  051106  046517  040440
```

```
 7587  047172  051104  020056  042522
 7588  047200  027107  000051
 7589
 7590  047204  020040  042524  052123  DH66:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>'READ.'
 7591  047212  004456  040503  046114
 7592  047220  040440  020124  041520
 7593  047226  004456  051127  052117
 7594  047234  027105  051011  040505
 7595  047242  027104
 7596  047244  042411  050130  041505          .ASCIZ  <TAB>'EXPECTED.'
 7597  047252  042524  027104     000
 7598
 7599  047257     040  052040  051505  DH67:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN READ FROM THE '
 7600  047264  027124  041411  046101
 7601  047272  020114  052101  050040
 7602  047300  027103  050011  052101
 7603  047306  042524  047122  051040
 7604  047314  040505  020104  051106
 7605  047322  046517  052040  042510
 7606  047330     040
 7607  047331     110  052111  046457          .ASCIZ  'HIT/MISS REGISTER.'
 7608  047336  051511  020123  042522
 7609  047344  044507  052123  051105
 7610  047352  000056
 7611
 7612          047257                  DH70=DH67
 7613
 7614          047257                  DH71=DH67
 7615
 7616          047257                  DH72=DH67
 7617
 7618          047257                  DH73=DH67
 7619
 7620          047257                  DH74=DH67
 7621
 7622  047354  020040  042524  052123  DH75:   .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>' GROUP.'<TAB>
 7623  047362  004456  040503  046114
 7624  047370  040440  020124  041520
 7625  047376  004456  043440  047522
 7626  047404  050125  004456
 7627  047410  042101  051104  051505          .ASCIZ  'ADDRESS.'<TAB>'PATTERN IN CONTROL REG.'
 7628  047416  027123  050011  052101
 7629  047424  042524  047122  044440
 7630  047432  020116  047503  052116
 7631  047440  047522  020114  042522
 7632  047446  027107     000
 7633
 7634          047354                  DH76=DH75
 7635
 7636  047451     040  052040  051505  DH77:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
 7637  047456  027124  041411  046101
 7638  047464  020114  052101  050040
 7639  047472  027103     000
 7640
 7641
 7642          047354                  DH117=DH75
```

```
7643
7644  047475    040  052040  051505  DH12C:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN IN CONTROL REG.'
7645  047502  027124  041411  046101
7646  047510  020114  052101  050040
7647  047516  027103  050011  052101
7648  047524  042524  047122  044440
7649  047532  020116  047503  052116
7650  047540  047522  020114  042522
7651  047546  027107     000
7652
7653  047551    040  052040  051505  DH121:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TEST ADDRESS.'
7654  047556  027124  041411  046101
7655  047564  020114  052101  050040
7656  047572  027103  052011  051505
7657  047600  020124  042101  051104
7658  047606  051505  027123     000
7659
7660  047613    040  052040  051505  DH122:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>
7661  047620  027124  041411  046101
7662  047626  020114  052101  050040
7663  047634  027103  053411  047522
7664  047642  042524  004456
7665  047646  044124  047105  041440          .ASCIZ  'THEN CLEARED AND READ.'
7666  047654  042514  051101  042105
7667  047662  040440  042116  051040
7668  047670  040505  027104     000
7669
7670  047675    040  042524  052123  DH123:  .ASCIZ  ' TEST.'<TAB>'CALL AT PC.'<TAB>'WROTE.'<TAB>'READ.'
7671  047702  004456  040503  046114
7672  047710  040440  020124  041520
7673  047716  004456  051127  052117
7674  047724  027105  051011  040505
7675  047732  027104     000
7676
7677          046531                  DH124=DH14
7678
7679  047735    040  052040  051505  DH125:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ADDRESS.'
7680  047742  027124  041411  046101
7681  047750  020114  052101  050040
7682  047756  027103  040411  042104
7683  047764  042522  051523  000056
7684
7685  047772  020040  042524  052123  DH126:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'
7686  050000  004456  040503  046114
7687  050006  040440  020124  041520
7688  050014  004456  051124  050101
7689  050022  040440  020124  041520
7690  050030     056
7691  050031     011  051105  047522          .ASCIZ  <TAB>'ERROR REG.'
7692  050036  020122  042522  027107
7693  050044     000
7694
7695  050045    040  052040  051505  DH127:  .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'PATTERN USED.'
7696  050052  027124  041411  046101
7697  050060  020114  052101  050040
7698  050066  027103  050011  052101
```

```
 7699  050074  042524  047122  052440
 7700  050102  042523  027104     000
 7701
 7702  050107     040  052040  051505  DH130:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR ADR REG.'
 7703  050114  027124  041411  046101
 7704  050122  020114  052101  050040
 7705  050130  027103  042411  051122
 7706  050136  051117  040440  051104
 7707  050144  051040  043505     056
 7708  050151     011  051105  047522          .ASCIZ  <TAB>'ERROR REG.'
 7709  050156  020122  042522  027107
 7710  050164     000
 7711
 7712  050165     040  052040  051505  DH131:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
 7713  050172  027124  041411  046101
 7714  050200  020114  052101  050040
 7715  050206  027103  052011  040522
 7716  050214  020120  052101  050040
 7717  050222  027103     011
 7718  050225     105  051122  051117          .ASCIZ  'ERROR ADR REG.'
 7719  050232  040440  051104  051040
 7720  050240  043505  000056
 7721
 7722          047735                  DH132=DH125
 7723
 7724          047772                  DH133=DH126
 7725
 7726  050244  020040  042524  052123  DH134:  .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'TRAP AT PC.'<TAB>
 7727  050252  004456  040503  046114
 7728  050260  040440  020124  041520
 7729  050266  004456  051124  050101
 7730  050274  040440  020124  041520
 7731  050302  004456
 7732  050304  047503  052116  047522          .ASCIZ  'CONTROL REG.'
 7733  050312  020114  042522  027107
 7734  050320     000
 7735
 7736          047451                  DH135=DH77
 7737
 7738  050321     040  052040  051505  DH150:  .ASCIZ  '  TEST.'<TAB>'TRAP AT PC.'<TAB>'CALL AT PC.'<TAB>'CPU ERROR REGISTER.'
 7739  050326  027124  052011  040522
 7740  050334  020120  052101  050040
 7741  050342  027103  041411  046101
 7742  050350  020114  052101  050040
 7743  050356  027103  041411  052520
 7744  050364  042440  051122  051117
 7745  050372  051040  043505  051511
 7746  050400  042524  027122     000
 7747                                  ;THESE ARE DATA FORMAT DESIGNATORS FOR THE DATA TABLE:
 7748
 7749  050405     004     004     003  DF1:    .BYTE   4,4,3,3
 7750  050410     003
 7751
 7752  050411     004     003     007  DF14:   .BYTE   4,3,7,3,0
 7753  050414     003     000
 7754
```

```
 7755  050416    004     003          DF15:   .BYTE   4,3
 7756
 7757  050420    004     003     003  DF55:   .BYTE   4,3,3,2
 7758  050423    002
 7759
 7760            050420               DF56=DF55
 7761
 7762            050420               DF57=DF55
 7763
 7764            050420               DF60=DF55
 7765
 7766            050420               DF61=DF55
 7767
 7768            050420               DF62=DF55
 7769
 7770  050424    004     003     000  DF63:   .BYTE   4,3,0,0,0
 7771  050427    000     000
 7772
 7773            050424               DF64=DF63
 7774
 7775            050424               DF65=DF63
 7776
 7777            050424               DF66=DF63
 7778
 7779            050424               DF67=DF63
 7780
 7781            050424               DF70=DF63
 7782
 7783            050424               DF71=DF63
 7784
 7785            050424               DF72=DF63
 7786
 7787            050424               DF73=DF63
 7788
 7789            050424               DF74=DF63
 7790
 7791  050431    004     003     004  DF75:   .BYTE   4,3,4,2,0
 7792  050434    002     000
 7793
 7794            050431               DF76=DF75
 7795
 7796  050436    004     003     005  DF77:   .BYTE   4,3,5,2,5,0,5,2,5,0
 7797  050441    002     005     000
 7798  050444    005     002     005
 7799  050447    000
 7800
 7801
 7802            050431               DF117=DF75
 7803
 7804  050450    004     003     000  DF120:  .BYTE   4,3,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0,5,0
 7805  050453    005     000     005
 7806  050456    000     005     000
 7807  050461    005     000     005
 7808  050464    000     005     000
 7809  050467    005     000     005
 7810  050472    000     005     000
```

```
 7811  050475  005     000     005
 7812  050500  000     005     000
 7813
 7814  050503  004     003     002  DF121:  .BYTE   4,3,2,2
 7815  050506  002
 7816
 7817  050507  004     003     000  DF122:  .BYTE   4,3,0,0
 7818  050512  000
 7819
 7820          050507                       DF123=DF122
 7821
 7822  050513  004     003     007  DF124:  .BYTE   4,3,7,3,0,5,0,
 7823  050516  003     000     005
 7824  050521  000     000
 7825
 7826  050523  004     003     002  DF125:  .BYTE   4,3,2,0
 7827  050526  000
 7828
 7829  050527  004     003     003  DF126:  .BYTE   4,3,3,0,5,2,5,2
 7830  050532  000     005     002
 7831  050535  005     002
 7832
 7833  050537  004     003     000  DF127:  .BYTE   4,3,0
 7834
 7835          050523                       DF130=DF125
 7836
 7837  050542  004     003     003  DF131:  .BYTE   4,3,3,2,5,0,5,0,5,0
 7838  050545  002     005     000
 7839  050550  005     000     005
 7840  050553  000
 7841
 7842          050523                       DF132=DF125
 7843
 7844          050527                       DF133=DF126
 7845
 7846  050554  004     003     003  DF134:  .BYTE   4,3,3,0,5,2,0
 7847  050557  000     005     002
 7848  050562  000
 7849
 7850  050563  004     003     005  DF135:  .BYTE   4,3,5,0,5,0,5,2,5,2
 7851  050566  000     005     000
 7852  050571  005     002     005
 7853  050574  002
 7854
 7855  050575  004     003     003  DF150:  .BYTE   4,3,3,0
 7856  050600  000
 7857
 7858          050602                       .EVEN
 7859
 7860                                       ;THESE ARE DATA TABLES:
 7861
 7862  050602  001232  001234  001236  DT1:    .WORD   $TMP0,$TMP1,$TMP2,$ERRPC,0
 7863  050610  001116  000000
 7864
 7865  050614  001232  001116  001234  DT14:   .WORD   $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,0
 7866  050622  001240  001242  000000
```

```
7867
7868  050630  001232  001234  000000  DT15:   .WORD   $TMP0,$TMP1,0
7869
7870
7871  050636  001232  001234  001116  DT55:   .WORD   $TMP0,$TMP1,$ERRPC,$TMP3,0
7872  050644  001240  000000
7873
7874          050636                  DT56=DT55
7875
7876          050636                  DT57=DT55
7877
7878          050636                  DT60=DT55
7879
7880          050636                  DT61=DT55
7881
7882          050636                  DT62=DT55
7883
7884  050650  001232  001116  001236  DT63:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
7885  050656  001240  000000
7886
7887  050662  001232  001116  001236  DT64:   .WORD   $TMP0,$ERRPC,$TMP2,0
7888  050670  000000
7889
7890  050672  001232  001116  001236  DT65:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
7891  050700  001240  000000
7892
7893  050704  001232  001116  001236  DT66:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,$TMP4,0
7894  050712  001240  001242  000000
7895
7896          050662                  DT67=DT64
7897
7898          050662                  DT70=DT64
7899
7900          050662                  DT71=DT64
7901
7902          050662                  DT72=DT64
7903
7904          050662                  DT73=DT64
7905
7906          050662                  DT74=DT64
7907
7908  050720  001232  001116  001236  DT75:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP10,$TMP3,0
7909  050726  001252  001240  000000
7910
7911  050734  001232  001116  001236  DT76:   .WORD   $TMP0,$ERRPC,$TMP2,$TMP12,$TMP3,0
7912  050742  001256  001240  000000
7913
7914  050750  001232  001116  033634  DT77:   .WORD   $TMP0,$ERRPC,MTA77,$TMP10,MTB77,$TMP2,MTC77
7915  050756  001252  033650  001236
7916  050764  033712
7917  050766  001256  033747  001240          .WORD   $TMP12,MTD77,$TMP3,0
7918  050774  000000
7919
7920          050734                  DT117=DT76
7921
7922  050776  001232  001116  001236  DT120:  .WORD   $TMP0,$ERRPC,$TMP2,MTA120,KCR0,MTG120,KCE0
```

```
7923  051004  034117  006770  034337
7924  051012  007004
7925  051014  034147  006772  034337          .WORD   MTB120,KCR1,MTG120,KCE1
7926  051022  007006
7927  051024  034177  006774  034337          .WORD   MTC120,KCR2,MTG120,KCE2
7928  051032  007010
7929  051034  034227  006776  034337          .WORD   MTD120,KCR3,MTG120,KCE3
7930  051042  007012
7931  051044  034257  007000  034337          .WORD   MTE120,KCR4,MTG120,KCE4
7932  051052  007014
7933  051054  034307  007002  034337          .WORD   MTF120,KCR5,MTG120,KCE5,0
7934  051062  007016  000000
7935
7936  051066  001232  001116  001236  DT121:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP4,0
7937  051074  001242  000000
7938
7939  051100  001232  001116  001236  DT122:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,0
7940  051106  001240  000000
7941
7942          051100                  DT123=DT122
7943
7944  051112  001232  001116  001234  DT124:  .WORD   $TMP0,$ERRPC,$TMP1,$TMP3,$TMP4,MTA124,$TMP6,0
7945  051120  001240  001242  034400
7946  051126  001246  000000
7947
7948  051132  001232  001116  001236  DT125:  .WORD   $TMP0,$ERRPC,$TMP2,0
7949  051140  000000
7950
7951  051142  001232  001116  001236  DT126:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP7,MTA126,$TMP5,MTB126,$TMP3,0
7952  051150  001250  034472  001244
7953  051156  034520  001240  000000
7954
7955          051132                  DT127=DT125
7956
7957  051164  001232  001116  001236  DT130:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP4,0
7958  051172  001242  000000
7959
7960  051176  001232  001116  001236  DT131:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA131,$TMP5
7961  051204  001240  034552  001244
7962  051212  034634  001246  034667          .WORD   MTB131,$TMP6,MTC131,$TMP7,0
7963  051220  001250  000000
7964
7965          051132                  DT132=DT125
7966
7967          051142                  DT133=DT126
7968
7969  051224  001232  001116  001236  DT134:  .WORD   $TMP0,$ERRPC,$TMP2,$TMP3,MTA134,$TMP4,$TMP6,0
7970  051232  001240  034715  001242
7971  051240  001246  000000
7972
7973  051244  001232  001116  034751  DT135:  .WORD   $TMP0,$ERRPC,MTA135,$TMP2,MTB135,$TMP3
7974  051252  001236  035001  001240
7975  051260  035023  001242  035057          .WORD   MTC135,$TMP4,MTD135,$TMP6,0
7976  051266  001246  000000
7977
7978  051272  001232  001234  001236  DT150:  .WORD   $TMP0,$TMP1,$TMP2,$TMP3,0
```

```
 7979  051300  001240  000000
 7980
 7981  051304  000000  000000  000000  BOTTOM: .WORD   0,0,0
 7982          057312                          .=.+^D200
 7983  057312                                  BOTPRG:
 7984          000001                          .END
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
ABORTT  030620          5765    5999#
ADRNG   033111          5934    6360#
BACKAD  030736          5904*   5913    5914#
BIT0   = 000001          195#   1301    1399    4349    4422    4487    5093    5214
BIT00  = 000001          175#    185
BIT01  = 000002          174#    184
BIT02  = 000004          173#    183
BIT03  = 000010          172#    182
BIT04  = 000020          171#    181
BIT05  = 000040          170#    180
BIT06  = 000100          169#    179
BIT07  = 000200          168#    178
BIT08  = 000400          167#    177    5376
BIT09  = 001000          166#    176    5386    5445
BIT1   = 000002          184#   1306    1404
BIT10  = 002000          165#   5430
BIT11  = 004000          164#   5393
BIT12  = 010000          163#
BIT13  = 020000          162#   5437
BIT14  = 040000          161#   5362
BIT15  = 100000          160#
BIT2   = 000004          183#   1311    1409
BIT3   = 000010          182#
BIT4   = 000020          181#
BIT5   = 000040          180#
BIT6   = 000100          179#    999    6028
BIT7   = 000200          178#   6008
BIT8   = 000400          177#
BIT9   = 001000          176#
BOTPRG  057312          7983#
BOTTOM  051304           989    6014    7981#
BPTVEC= 000014            192#
CACHVE= 000114            200#   1002*   1019*   1111*   1443*   1536*   1627*   1838*   1949*   2017*   2031*   2058*   2134*
                         2219*   2303*   2390*   2471*   2566*   2654*   2753*   2852*   2951*   3050*   3149*   3248*   3347*
                         3450*   3574*   3705*   3817*   3928*   4039*   4150*   4235*   4349*   4423*   4488*   4548*   4572*
                         4686*   4712*   4828*   4853*   4969*   4995*   5086*   5097*   5119*   5207*   5218*   5240*   5868*
                         5874*   5902*
CHAINQ  031156          5331    6013#
CLEAN   030650          5764    5900#
CNRNG   033321          5948    6387#
CONCMS  032064          6012    6240#
CONFLG  031064          1074*   1130*   1137*   1229*   1812*   5944    5970#
CONFL2  031100          1322*   1418*   1481*   1504*   1574*   1597*   1785*   1810    5976#
CONTRL= 177746           209#   1072    1116*   1118    1211    1254*   1269*   1292*   1313*   1351*   1367*   1380*   1410*
                         1459*   1472*   1483*   1496*   1506*   1552*   1565*   1576*   1589*   1599*   1642*   1644*   1651*
                         1789*   1852*   1870*   1886*   1957*   2030*   2062*   2137*   2222*   2306*   2393*   2474*   2567*
                         2657*   2756*   2855*   2954*   3053*   3152*   3251*   3350*   3483*   3599*   3706*   3818*   3929*
                         4040*   4151*   4347*   4348*   4375    4407*   4422*   4440    4472*   4487*   4505    4545*   4683*
                         4823*   4964*   5088*   5209*   5909*
CPSPUR  030474          1001    3451    4285    5860#   5903
CPUERR= 177766           221#   1044    1092*   1095*   4283    5457*   5862    5911*
CR     = 000015           97#   5566    5576
CRLF   = 000200           98#    974    5537    5576    6240    6242    6249    6259    6262    6269    6279    6289    6292
                         6302    6308    6311    6316    6323    6328    6335    6345    6348    6355    6360    6374    6387
                         6399    6413    6426    6436    6449    6454    6462    6468    6474    6479    6485    6491    6504
                         6516    6521    6527    6537    6543    6548    6554    6565    6590    6604    6623    6642    6659
```

```
                    6674    6691    6698    6717    6724    6743    6751    6757    6775    6783    6821    6829    6846
                    6854    6877    6885    6917    6934    6943    6967    6976    6992    7012    7030    7054    7083
                    7105    7122    7145    7161    7200    7206    7227    7234    7247    7269    7276    7284    7311
                    7318    7332    7354    7361    7374    7429    7443    7466    7484    7499
DDISP = 177570        91#     529     954
DF1     050405       607    7749#
DF117 = 050431       845    7802#
DF120   050450       848    7804#
DF121   050503       851    7814#
DF122   050507       854    7817#   7920
DF123 = 050507       857    7820#
DF124   050513       860    7822#
DF125   050523      7826#   7835    7942
DF126   050527      7829#   7844
DF127   050537       869    7833#
DF130 = 050523       872    7835#
DF131   050542       876    7837#
DF132 = 050523       879    7842#
DF133 = 050527       882    7844#
DF134   050554       885    7846#
DF135   050563       888    7850#
DF14    050411       640    7752#
DF140   045511       897    7400#   7403    7405    7407
DF141 = 045511       900    7403#
DF142 = 045511       903    7405#
DF143 = 045511       906    7407#
DF15    050416       643    7755#
DF150   050575       921    7855#
DF55    050420       741    7757#   7760    7762    7764    7766    7768
DF56  = 050420       744    7760#
DF57  = 050420       747    7762#
DF60  = 050420       750    7764#
DF61  = 050420       753    7766#
DF62  = 050420       756    7768#
DF63    050424       759    7770#   7773    7775    7777    7779    7781    7783    7785    7787    7789
DF64  = 050424       762    7773#
DF65  = 050424       765    7775#
DF66  = 050424       768    7777#
DF67  = 050424       771    7779#
DF70  = 050424       774    7781#
DF71  = 050424       777    7783#
DF72  = 050424       780    7785#
DF73  = 050424       783    7787#
DF74  = 050424       786    7789#
DF75    050431       790    7791#   7794    7802
DF76  = 050431       793    7794#
DF77    050436       796    7796#
DH1     046456       607    7511#
DH117 = 047354       845    7642#
DH120   047475       848    7644#
DH121   047551       851    7653#
DH122   047613       854    7660#
DH123   047675       857    7670#
DH124 = 046531       860    7677#
DH125   047735      7679#   7722
DH126   047772      7685#   7724
```

```
DH127    050045            869    7695#
DH130    050107            872    7702#
DH131    050165            876    7712#
DH132 =  047735            879    7722#
DH133 =  047772            882    7724#
DH134    050244            885    7726#
DH135 =  047451            888    7736#
DH14     046531            640    7519#  7677
DH140    045446            897    7387#  7394    7396    7398
DH141 =  045446            900    7394#
DH142 =  045446            903    7396#
DH143 =  045446            906    7398#
DH15     046624            643    7531#
DH150    050321            921    7738#
DH55     046650            741    7536#  7545    7547    7549    7551    7553
DH56  =  046650            744    7545#
DH57  =  046650            747    7547#
DH60  =  046650            750    7549#
DH61  =  046650            753    7551#
DH62  =  046650            756    7553#
DH63     046725            759    7555#
DH64     047027            762    7568#
DH65     047102            765    7577#
DH66     047204            768    7590#
DH57     047257            771    7599#  7612    7614    7616    7618    7620
DH70  =  047257            774    7612#
DH71  =  047257            777    7614#
DH72  =  047257            780    7616#
DH73  =  047257            783    7618#
DH74  =  047257            786    7620#
DH75     047354            790    7622#  7634    7642
DH76  =  047354            793    7634#
DH77     047451            796    7636#  7736
DISPLA   001142            529#    954*   962*   5407*   5429*
DISPRE   000174            485#    962
DSWR  =  177570             90#    528    953
DT1      050602            607    7862#
DT117 =  050734            845    7920#
DT120    050776            848    7922#
DT121    051066            851    7936#
DT122    051100            854    7939#  7942
DT123 =  051100            857    7942#
DT124    051112            860    7944#
DT125    051132           7948#   7955   7965
DT126    051142           7951#   7967
DT127 =  051132            869    7955#
DT130    051164            872    7957#
DT131    051176            876    7960#
DT132 =  051132            879    7965#
DT133 =  051142            882    7967#
DT134    051224            885    7969#
DT135    051244            888    7973#
DT14     050614            640    7865#
DT140    045516            897    7410#  7413    7415    7417
DT141 =  045516            900    7413#
DT142 =  045516            903    7415#
```

```
DT143 = 045516          906     7417#
DT15    050630          643     7868#
DT150   051272          921     7978#
DT55    050636          741     7871#   7874    7876    7878    7880    7882
DT56  = 050636          744     7874#
DT57  = 050636          747     7876#
DT60  = 050636          750     7878#
DT61  = 050636          753     7880#
DT62  = 050636          756     7882#
DT63    050650          759     7884#
DT64    050662          762     7887#   7896    7898    7900    7902    7904    7906
DT65    050672          765     7890#
DT66    050704          768     7893#
DT67  = 050662          771     7896#
DT70  = 050662          774     7898#
DT71  = 050662          777     7900#
DT72  = 050662          780     7902#
DT73  = 050662          783     7904#
DT74  = 050662          786     7906#
DT75    050720          790     7908#
DT76    050734          793     7911#   7920
DT77    050750          796     7914#
EMTVEC= 000030          195#     937*    938*
EM1     035105          607     6579#
EM117   041657          845     7030#
EM120   042006          848     7047#
EM121   042221          851     7073#
EM122   042422          854     7097#
EM123   042552          857     7113#
EM124   042753          860     7137#
EM127   043161          869     7161#
EM130   043343          872     7182#
EM131   043415          876     7191#
EM132   045530          879     7420#
EM133   045667          882     7438#
EM134   046041          885     7458#
EM135   046207          888     7477#
EM14    035172          640     6590#
EM140   043642          897     7218#
EM141   044203          900     7260#
EM142   044543          903     7302#
EM143   045105          906     7345#
EM15    035231          643     6597#
EM150   046372          921     7499#
EM55    035301          741     6604#
EM56    035445          744     6623#
EM57    035612          747     6642#
EM60    035734          750     6658#
EM61    036060          753     6674#
EM62    036210          756     6691#
EM63    036336          759     6708#
EM64    036555          762     6734#
EM65    036752          765     6757#
EM66    037335          768     6802#
EM67    037417          771     6812#
EM70    037634          774     6838#
```

```
EM71    040118          777     6869#
EM72    040370          780     6900#
EM73    040612          783     6926#
EM74    041076          786     6959#
EM75    041362          790     6992#   7010
EM76  = 041362          793     7010#
EM77    041521          796     7012#
ERRNG   033221         5941     6374#
ERRVEC= 000004          188#     951     952#    963#    1001#   1024    1025#   1038#   1046#   1047    3451#   4262#   5367
                       5368#   5370#   5373#   5903#
ERTYPE  031352         5439     5891    6085#
ERT1    031464         6115#    6179
ERT2    031676         6119     6124    6134    6142    6148    6160    6169    6173#
ERT3    031702         6154     6176#
ERT4    031712         6113     6178    6181#
ERT5    031714         6093     6182#
GNS   = ****** U        484      973    5754    5755    5756    5757    5758    5761    5762    5764    5765    5766    5767
                       5768     5769    5770    5771    5772
HIADRS= 177742          207#    1060    1170    2022    2086    2094    2102    2171    2180    2188    2255    2264    2272
                       2342    2351    2359    2428    2437    2445    2516    2525    2533    2602    2611    2619    2701
                       2710    2718    2800    2809    2817    2899    2908    2916    2998    3007    3015    3097    3106
                       3114    3196    3205    3213    3295    3304    3312    3394    3403    3411    3519    3537    3545
                       3631    3640    3648    3743    3752    3760    3854    3863    3871    3965    3974    3982    4076
                       4085    4093    4187    4196    4204    4294    4312    4320    4377    4442    4507    4608    4620
                       4629    4748    4760    4769    4889    4901    4910    5031    5043    5052    5877
HIAFLG  031060         1062*   1185*   5930    5968#
HIAFL2  031074         5974#
HIMFLG  031070         1086*   1413*   1815*   5958    5972#
HIMFL2  031104         1316*   1786*   1813    5978#
HITMIS= 177752          211#    1035    1084    1265    1278    1291    1362    1376    1389    1462    1475    1486    1498
                       1555    1568    1579    1591    1663    1860    1876    1906    2665    2764    2863    2962    3061
                       3160    3259    3358    4413    4478
HMRNG   033531         5962    6413#
HT    = 000011           95#   5535    5576
IOTVEC= 000020          193#     935*    936*
JA    = 000001         1014#
JADONE  004040         1039    1095#
JAERR   003616         1025    1044#
JAERR0  003626         1046#   1049
JAERR1  003640         1045    1048#
JAERR2  003706         1055    1060#
JAERR3  003726         1061    1066#
JAERR4  003746         1067    1072#
JAERR5  003766         1073    1078#
JAERR6  004006         1079    1084#
JAERR7  004026         1085    1090#
JAERR9  004030         1058    1064    1070    1076    1082    1088    1092#
JATMP   003614         1024*   1038    1041#   1046
JA1     003564         1028    1030#   1036
JA2     003570         1034#   1048    1093
JA3     003602         1038#
JB    = 000002         1106#
JBDONE  004200         1120    1132    1139#
JBERR1  004134         1126#
JBERR2  004164         1124    1134#
JB1     004102         1115    1116#
```

```
JB2      004120            1121#
JC     = 000003            1156#
JCDONE   004342            1174    1184    1187#
JCERR1   004272            1172    1176#
JC1      004244            1166    1168#
JC2      004264            1173#
JD     = 000004            1143    1202#
JDDONE   004460            1223    1231#
JDERR1   004432            1220    1224#
JD1      004404            1213    1214#   1221
JD2      004424            1221#   1230
KA     = 000006            1341#
KADONE   005320            1392    1410#
KAD2     005344            1412    1416#
KAD3     005362            1414    1417    1419#
KAERR1   005250            1364    1396#
KAERR2   005266            1378    1401#
KAERR3   005304            1391    1406#
KAFLG    005246            1350*   1394#   1399*   1404*   1409*   1411    1416
KA1      005046            1351#   1352
KA2      005070            1353    1356#   1357
KA3      005116            1366#   1400
KA4      005142            1368    1370#   1371
KA5      005170            1379#   1405
KA6      005214            1381    1383#   1384
KB     = 000005            1244#   1793
KBDONE   004754            1293    1313#
KBD2     004776            1315    1320#
KBD3     005012            1321    1324#
KBERR1   004704            1266    1298#
KBERR2   004722            1279    1303#
KBERR3   004740            1292    1308#
KBFLG    004702            1253*   1296#   1301*   1306*   1311*   1314    1320
KB1      004514            1254#   1255
KB2      004540            1257    1259#   1260
KB3      004562            1268#   1302
KB4      004606            1270    1272#   1273
KB5      004630            1281#   1307
KB6      004654            1283    1285#   1286
KC     = 000011            1622#
KCCON    006736            1631*   1651    1741*   1744*   1750#   1783
KCDONE   007050            1738    1789#
KCERR    007020            1728    1782#
KCE0     007004            1717    1724    1775#   7922
KCE1     007006            1776#   7925
KCE2     007010            1777#   7927
KCE3     007012            1778#   7929
KCE4     007014            1779#   7931
KCE5     007016            1780#   7933
KCFLG1   006740            1632*   1736*   1752#
KCPTR    006742            1634*   1648    1714    1731*   1732    1754#
KCR0     006770            1706*   1723    1768#   7922
KCR1     006772            1708*   1769#   7925
KCR2     006774            1709*   1770#   7927
KCR3     006776            1710*   1771#   7929
KCR4     007000            1711*   1772#   7931
```

```
KCR5    007002          1712*   1773#   7933
KCTBL   006744          1634    1757#
KCTBL8  006766          1732    1766#
KC0     006312          1633#   1742    1747
KC1     006326          1633    1639#   1734
KC10    006534          1705#
KC11    006574          1714#
KC12    006612          1718#   1721
KC13    006642          1726#   1729
KC14    006652          1727    1729#
KC15    006654          1731#   1787
KC16    006710          1737    1740#
KC17    006724          1740    1744#
KC2     006410          1653#   1661
KC2.5   006416          1652    1654    1656#
KC3     006452          1649    1664    1678#
KC4     006462          1679    1682#
KC5     006472          1683    1686#
KC6     006502          1687    1690#
KC7     006512          1691    1694#
KC8     006522          1695    1698#
KC9     006532          1699    1702#
KD    = 000007          1438#
KDPAR0= 172360           363#
KDPAR1= 172362           364#
KDPAR2= 172364           365#
KDPAR3= 172366           366#
KDPAR4= 172370           367#
KDPAR5= 172372           368#
KDPAR6= 172374           369#
KDPAR7= 172376           370#
KDPDR0= 172320           341#
KDPDR1= 172322           342#
KDPDR2= 172324           343#
KDPDR3= 172326           344#
KDPDR4= 172330           345#
KDPDR5= 172332           346#
KDPDR6= 172334           347#
KDPDR7= 172336           348#
KE    = 000010          1531#   1793
KERSTK= 001100            81#
KF    = 000013          1833#
KFTMP1  007336          1919#
KFTMP2  007340          1842    1920#
KF1     007144          1842#
KF2     007226          1861    1870#
KF3     007264          1877    1886#
KF4     007334          1909    1917#
KF5     007342          1917    1922#
KIPAR0= 172340           352#   3461    3576    3684    3796    3907    4018    4129    4243    4664    4804    4945    6216
KIPAR1= 172342           353#
KIPAR2= 172344           354#
KIPAR3= 172346           355#
KIPAR4= 172350           356#
KIPAR5= 172352           357#
KIPAR6= 172354           358#   4260*
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
KIPAR7= 172356            359#
KIPDR0= 172300            330#   3463    3578    3686    3798    3909    4020    4131    4245    4666    4806    4947
KIPDR1= 172302            331#
KIPDR2= 172304            332#
KIPDR3= 172306            333#
KIPDR4= 172310            334#
KIPDR5= 172312            335#
KIPDR6= 172314            336#
KIPDR7= 172316            337#
KSP   =%000006            116#
KTMP1D  005706           1509#
KTMP1E  006236           1602#
KTMP2D  005710           1448    1510#
KTMP2E  006240           1541    1603#
KV    = 000044           4337#
KVDONE  022506           4370    4381#
KVERR   022446           4349    4372#
KV1     022430           4352    4359#
KV2     022436           4362#
KX    = 000045           4397#
KXDONE  022706           4431    4447#
KXERR   022646           4423    4437#
KX1     022610           4414    4422#
KX2     022640           4408    4429#
KY    = 000012           1809#
KY1     007072           1811    1813#
KY2     007106           1814    1816#
KZ    = 000046           4462#
KZDONE  023106           4496    4512#
KZERR   023046           4488    4502#
KZ1     023010           4479    4487#
KZ2     023040           4473    4494#
K1D     005420           1448#
K1E     005750           1541#
K2D     005464           1459#
K2E     006014           1552#
K3D     005524           1463    1470#
K3E     006054           1556    1563#
K4D     005570           1476    1483#
K4E     006120           1569    1576#
K5D     005630           1487    1494#
K5E     006160           1580    1587#
K6D     005700           1493    1499    1506#
K6E     006230           1586    1592    1599#
K7D     005712           1507    1512#
K7E     006242           1600    1605#
LF    = 000012             96#   5570    5576
LKS   = 177546             92#
LKVEC = 000100            199#
LOADRS= 177740            206#   1027    1054    1169    2021    2085    2092    2101    2170    2178    2187    2254    2262
                         2271    2341    2349    2358    2427    2435    2444    2515    2523    2532    2601    2609    2618
                         2700    2708    2717    2799    2807    2816    2898    2906    2915    2997    3005    3014    3096
                         3104    3113    3195    3203    3212    3294    3302    3311    3393    3401    3410    3518    3536
                         3543    3630    3638    3647    3742    3750    3759    3853    3861    3870    3964    3972    3981
                         4075    4083    4092    4186    4194    4203    4293    4311    4318    4376    4441    4506    4599
                         4607    4619    4627    4739    4747    4759    4767    4880    4888    4900    4908    5022    5030
```

```
                    5042    5050    5155    5166    5276    5287    5872    5876
LOAFLG  031056      1020    1056*   1182*   5928    5967#
LOAFL2  031072      5973#
LOC     = 026200    1672#   1673#   1674#   1675    2140#   2141#   2142#   2143    2225#   2226#   2227#   2228    2309#
                    2310#   2311#   2312    2396#   2397#   2398#   2399    2484#   2485#   2486#   2487    2570#   2571#
                    2572#   2573    3602#   3603#   3604#   3605    3711#   3712#   3713#   3714    3823#   3824#   3825#
                    3826    3934#   3935#   3936#   3937    4045#   4046#   4047#   4048    4156#   4157#   4158#   4159
                    4354#   4355#   4356#   4357    4553#   4554#   4555#   4556    4577#   4578#   4579#   4580    4691#
                    4692#   4693#   4694    4717#   4718#   4719#   4720    4833#   4834#   4835#   4836    4858#   4859#
                    4860#   4861    4974#   4975#   4976#   4977    5000#   5001#   5002#   5003    5143#   5144#   5145#
                    5146    5264#   5265#   5266#   5267
LOOP    003364       983#   5341
MA      = 000014    1941#
MAB     = 000015    2048#
MABDON  010120      2074    2090    2096    2111#
MABRR0  007730      2058    2076#
MABRR1  007740      2079#
MABRR2  007756      2084#   2093    2095
MABRR3  010012      2082    2092#
MABRR4  010034      2077    2098#
MABR15  007742      2080#   2109
MAB2    007712      2070#
MADONE  007612      2008    2030#
MAERR   007526      1949    2010#
MAERR1  007550      2015    2020#
MAINT   = 177750     210#   1078    1121*   1122    1959    2061    2136    2221    2305    2392    2473    2565    2656
                    2755    2854    2953    3052    3151    3250    3349    3484    3598    3708    3820    3931    4042
                    4153    4351    4426    4491    4550    4574    4688    4714    4830    4855    4971    4997    5099
                    5122    5220    5243    5906#
MANFLG  031066      1080#   1131#   1995#   4730#   4846#   4987#   5013#   5951    5971#
MANFL2  031102      2004#   2026#   2073#   2107#   2156#   2194#   2240#   2278#   2327#   2365#   2413#   2451#   2501#
                    2539#   2507#   2625#   2686#   2724#   2785#   2823#   2884#   2922#   2983#   3021#   3082#   3120#
                    3181#   3219#   3280#   3318#   3379#   3417#   3503#   3524#   3616#   3654#   3729#   3766#   3839#
                    3877#   3950#   3988#   4061#   4099#   4172#   4210#   4566#   4590#   4704#   4871#   5977#

MAPH0   = 170202       446#
MAPH00= 170202       382#    446     4265#
MAPH01= 170206       384#    448
MAPH02= 170212       386#    450
MAPH03= 170216       388#    452
MAPH04= 170222       390#    454
MAPH05= 170226       392#    456
MAPH06= 170232       394#    458
MAPH07= 170236       396#    460
MAPH1   = 170206     448#
MAPH10= 170242       398#
MAPH11= 170246       400#
MAPH12= 170252       402#
MAPH13= 170256       404#
MAPH14= 170262       406#
MAPH15= 170266       408#
MAPH16= 170272       410#
MAPH17= 170276       412#
MAPH2   = 170212     450#
MAPH20= 170302       414#
MAPH21= 170306       416#
MAPH22= 170312       418#
```

```
MAPH23= 170316           420#
MAPH24= 170320           422#
MAPH25= 170326           424#
MAPH26= 170332           426#
MAPH27= 170336           428#
MAPH3 = 170216           462#
MAPH30= 170342           430#
MAPH31= 170346           432#
MAPH32= 170352           434#
MAPH33= 170356           436#
MAPH34= 170362           438#
MAPH35= 170366           440#
MAPH36= 170372           442#
MAPH37= 170376           444#
MAPH4 = 170222           454#
MAPH5 = 170226           456#
MAPH6 = 170232           458#
MAPH7 = 170236           460#
MAPL0 = 170200           445#
MAPL00= 170200           381#    445    3466    3581    3689    3801    3912    4023    4134    4264#    4669    4809    4950
MAPL01= 170204           383#    447
MAPL02= 170210           385#    449
MAPL03= 170214           387#    451
MAPL04= 170220           389#    453
MAPL05= 170224           391#    455
MAPL06= 170230           393#    457
MAPL07= 170234           395#    459
MAPL1 = 170204           447#
MAPL10= 170240           397#
MAPL11= 170244           399#
MAPL12= 170250           401#
MAPL13= 170254           403#
MAPL14= 170260           405#
MAPL15= 170264           407#
MAPL16= 170270           409#
MAPL17= 170274           411#
MAPL2 = 170210           449#
MAPL20= 170300           413#
MAPL21= 170304           415#
MAPL22= 170310           417#
MAPL23= 170314           419#
MAPL24= 170320           421#
MAPL25= 170324           423#
MAPL26= 170330           425#
MAPL27= 170334           427#
MAPL3 = 170214           451#
MAPL30= 170340           429#
MAPL31= 170344           431#
MAPL32= 170350           433#
MAPL33= 170354           435#
MAPL34= 170360           437#
MAPL35= 170364           439#
MAPL36= 170370           441#
MAPL37= 170374           443#
MAPL4 = 170220           453#
MAPL5 = 170224           455#
```

```
MAPL6 = 170230            457#
MAPL7 = 170234            459#
MA1     007426           1961    1964#   2007
MA2     007444           1986#
MA3     007472           1989    1997#
MA4     007516           1998    2006#   2028
MB    = 000016           2124#
MBDONE  010436           2157    2176    2182    2197#
MBERRO  010236           2134    2159#
MB1     010210           2138    2145#
MB2     010214           2147#
MB3     010220           2152#
MC    = 000017           2209#
MCDONE  010752           2241    2260    2266    2281#
MCERRO  010552           2219    2243#
MC1     010524           2223    2230#
MC2     010530           2232#
MC3     010534           2236#
MD    = 000020           2293#
MDDONE  011272           2328    2347    2353    2368#
MDERRO  011072           2303    2330#
MD1     011042           2307    2315#
MD2     011046           2317#
MD3     011054           2323#
ME    = 000021           2380#
MEDONE  011612           2414    2433    2439    2454#
MEERRO  011412           2390    2416#
MEMERR= 177744            208#   1066    1165*   2014    2020    2076    2080*   2081    2087    2105    2159    2165*   2166
                         2172    2191    2243    2249*   2250    2256    2275    2330    2336*   2337    2343    2362    2416
                         2422*   2423    2429    2448    2504    2510*   2511    2517    2536    2590    2596*   2597    2603
                         2622    2689    2695*   2696    2702    2721    2788    2794*   2795    2801    2820    2887    2893*
                         2894    2900    2919    2986    2992*   2993    2999    3018    3085    3091*   3092    3098    3117
                         3184    3190*   3191    3197    3216    3283    3289*   3290    3296    3315    3382    3388*   3389
                         3395    3414    3511    3522    3531*   3532    3538    3619    3625*   3626    3632    3651    3731
                         3737*   3738    3744    3763    3842    3848*   3849    3855    3874    3953    3959*   3960    3966
                         3985    4064    4070*   4071    4077    4096    4175    4181*   4182    4188    4207    4287    4297
                         4306*   4307    4313    4378    4443    4508    4597    4604    4614*   4615    4621    4737    4744
                         4754*   4755    4761    4878    4885    4895*   4896    4902    5020    5027    5037*   5038    5044
                         5152    5159*   5163    5170*   5273    5280*   5284    5291*   5456*   5869    5875    5910*
ME1     011362           2394    2402#
ME2     011366           2404#
ME3     011374           2409#
MF    = 000022           2466#
MFDONE  012126           2502    2521    2527    2542#
MFERRO  011726           2471    2504#
MF1     011674           2482    2489#
MF2     011706           2475    2495#
MF3     011710           2497#
MG    = 000023           2554#
MGDONE  012446           2588    2607    2613    2628#
MGERRO  012246           2566    2590#
MG1     012214           2568    2575#
MG2     012222           2578#
MG3     012230           2583#
MH    = 000024           2644#
MHDONE  013012           2687    2706    2712    2727#
```

DEKBC8.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
MHERRO  012612          2654    2689#
MH1     012572          2660    2676#
MH2     012574          2682#
MI    = 000025          2743#
MIDONE  013356          2786    2805    2811    2826#
MIERRO  013156          2753    2788#
MI1     013136          2759    2775#
MI2     013140          2781#
MJ    = 000026          2842#
MJDONE  013722          2885    2904    2910    2925#
MJERRO  013522          2852    2887#
MJ1     013502          2859    2874#
MJ2     013504          2880#
MK    = 000027          2941#
MKDONE  014266          2984    3003    3009    3024#
MKERRO  014066          2951    2986#
MK1     014046          2957    2973#
MK2     014050          2979#
ML    = 000030          3040#
MLDONE  014632          3083    3102    3108    3123#
MLERRO  014432          3050    3085#
ML1     014412          3056    3072#
ML2     014414          3078#
MMDES   031254          5766    6045#
MMESRS  032070          6024    6242#
MMRFLG  031062          1068*   2089*   2175*   2259*   2346*   2432*   2520*   2606*   2705*   2804*   2903*   3002*   3101*
                        3200*   3299*   3398*   3540*   3635*   3747*   3858*   3969*   4080*   4191*   4315*   4624*   4764*
                        4905*   5047*   5937    5969#
MMRFL2  031076          2108*   2195*   2279*   2366*   2452*   2540*   2626*   2725*   2824*   2923*   3022*   3121*   3220*
                        3319*   3418*   3655*   3767*   3878*   3989*   4100*   4211*   4299*   5975#
MMR0  = 177572           231#    235    2163*   2247*   2334*   2420*   2508*   2594*   2693*   2792*   2891*   2990*   3089*
                        3188*   3287*   3386*   3481*   3529*   3596*   3623*   3704*   3735*   3816*   3846*   3927*   3957*
                        4038*   4068*   4149*   4179*   4259*   4304*   4612*   4711*   4752*   4827*   4850*   4893*   4968*
                        4994*   5035*   5907*   6205
MMR1  = 177574           232#    236
MMR2  = 177576           233#    237
MMR3  = 172516           234#    238    2164*   2248*   2335*   2421*   2509*   2595*   2694*   2793*   2892*   2991*   3090*
                        3189*   3288*   3387*   3480*   3530*   3595*   3624*   3703*   3736*   3815*   3847*   3926*   3958*
                        4037*   4069*   4148*   4180*   4258*   4305*   4613*   4710*   4753*   4826*   4851*   4894*   4967*
                        4993*   5036*   5908*
MMSKIP= 104412          3449    3573    3682    3794    3905    4016    4127    4241    4662    4802    4943    5766#
MMVEC = 000250           202#
MN    = 000031          3139#
MNRNG   033423          5955    6399#
MO    = 000032          3238#
MODONE  015542          3281    3300    3306    3321#
MOERRO  015342          3248    3283#
MONF    031252           983*   6018*   6033#
MONTTY  031250           986*   6025    6031#
MO1     015322          3254    3270#
MO2     015324          3276#
MP    = 000033          3337#
MPDONE  016106          3380    3399    3405    3420#
MPERRO  015706          3347    3382#
MP1     015666          3353    3369#
MP2     015670          3375#
```

```
MQ     = 000043           4230#
MQDONE   022332           4281    4316    4323#
MQERR    022126           4262    4283#
MQ1      022106           4277#
MQ2      022142           4284    4287#
MQ3      022230           4288    4302#
MQ4      022232           4300    4304#
MQ5      022256           4310#   4319    4321
MQ6      022312           4308    4318#
MR     = 000034           3439#
MRDONE   016536           3504    3541    3548#
MRERRO   016330           3450    3506#
MR1      016310           3500#
MR2      016342           3508    3511#
MR3      016432           3512    3527#
MR4      016436           3525    3529#
MR5      016462           3535#   3544    3546
MR6      016516           3533    3543#
MS     = 000035           3563#
MSDONE   017156           3617    3636    3642    3657#
MSERRO   016756           3574    3619#
MSIZER   031276           5767    6061#
MS1      016730           3600    3607#
MS2      016734           3609#
MS3      016740           3612#
MT     = 000036           3672#
MTA101   034012           6449#
MTA11    032260           6269#
MTA120   034117           6462#   7922
MTA124   034400           6504#   7944
MTA126   034472           6516#   7951
MTA131   034552           6527#   7960
MTA134   034715           6548#   7969
MTA135   034751           6554#   7973
MTA17    032325           6277#   6300
MTA20    032361           6286#
MTA21    032370           6289#
MTA43    032455           6302#
MTA45    032530           6311#
MTA5     032176           6259#
MTA50    032606           6323#
MTA77    033634           6426#   7914
MTB120   034147           6468#   7925
MTB126   034520           6521#   7951
MTB131   034634           6537#   7962
MTB135   035001           6560#   7973
MTB17    032332           6279#
MTB21  = 032325           6300#
MTB45    032556           6316#
MTB77    033650           6429#   7914
MTC120   034177           6474#   7927
MTC131   034667           6543#   7962
MTC135   035023           6565#   7975
MTC17    032352           6283#
MTC45    032573           6320#
MTC77    033712           6436#   7914
```

```
MTDONE  017602          3729    3748    3754    3769#
MTD120  034227          6479#   7929
MTD135  035057          6571#   7975
MTD77   033747          6442#   7917
MTERRO  017402          3705    3731#
MTE120  034257          6485#   7931
MTF120  034307          6491#   7933
MTG120  034337          6497#   7922    7925    7927    7929    7931    7933
MT1     017352          3709    3717#
MT2     017364          3724#
MU    = 000037          3784#
MUDONE  020222          3840    3859    3865    3880#
MUERRO  020022          3817    3842#
MU1     017776          3821    3829#
MU2     020004          3835#
MV    = 000040          3895#
MVDONE  020642          3951    3970    3976    3991#
MVERRO  020442          3928    3953#
MV1     020416          3932    3940#
MV2     020424          3946#
MW    = 000041          4006#
MWDONE  021262          4062    4081    4087    4102#
MWERRO  021062          4039    4064#
MW1     021036          4043    4051#
MW2     021044          4057#
MX    = 000042          4117#
MXDONE  021702          4173    4192    4198    4213#
MXERRO  021502          4150    4175#
MX1     021456          4154    4162#
MX2     021464          4168#
MOM1  = 000014           474#   1254    1957    2062    2137    2222    2306    2393    2474    2567    3483    3705    4347
                        4545    4683    4823    4964    5088    5209
M1MO  = 000014           473#    474    3599
NA    = 000047          4535#
NADONE  023472          4567    4591    4625    4632#
NA1     023174          4551    4558#
NA10    023452          4616    4627#
NA2     023200          4560#   4599    4605
NA3     023226          4548    4570#
NA4     023250          4575    4582#
NA5     023254          4584#
NA6     023302          4572    4594#
NA7     023326          4598    4602#
NA8     023372          4600    4612#
NA9     023416          4618#   4628    4630
NB    = 000050          4652#
NBDONE  024162          4705    4731    4765    4772#
NB1     023650          4689    4696#
NB10    024142          4756    4767#
NB2     023654          4698#   4739    4745
NB3     023702          4686    4708#
NB4     023740          4715    4722#
NB5     023744          4724#
NB6     023772          4712    4734#
NB7     024016          4738    4742#
NB8     024062          4740    4752#
```

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
NB9     024106          4758#   4768    4770
NC    = 000051          4792#
NCDONE  024662          4847    4872    4906    4913#
NC1     024354          4831    4838#
NC10    024642          4897    4908#
NC2     024360          4840#   4880    4896
NC3     024406          4828    4850#
NC4     024440          4856    4863#
NC5     024444          4865#
NC6     024472          4853    4875#
NC7     024516          4879    4883#
NC8     024562          4881    4893#
NC9     024606          4899#   4909    4911
ND    = 000052          4933#
NDDONE  025366          4989    5014    5048    5055#
ND1     025054          4972    4979#
ND10    025346          5039    5050#
ND2     025060          4981#   5022    5028
ND3     025106          4969    4991#
ND4     025144          4998    5005#
ND5     025150          5007#
ND6     025176          4995    5017#
ND7     025222          5021    5025#
ND8     025266          5023    5035#
ND9     025312          5041#   5051    5053
NMDONE  015176          3182    3201    3207    3222#
NMERRO  014776          3149    3184#
NM1     014756          3155    3171#
NM2     014760          3177#
NOCNC   031230          6010    6027#
PARCNT  031106          5092    5213    5988#
PC    =%000007           119#   5092*   5213*   5318*   5321*   5331*   5335*   5340    5439*   5544*   5551*   5558*   5572*
                        5574*   5810    5844*   5891*   5995*   6022*   6129*   6147*   6159*   6165*   6183*   6228*   6236*
PDMSG1  032634          6328#
PDMSG2  033012          6348#
PIRQ  = 177772            89#
PIRQVE= 000240           201#
POWERM  032127          5809    6249#
PR0   = 000000           122#
PR1   = 000040           123#
PR2   = 000100           124#
PR3   = 000140           125#
PR4   = 000200           126#
PR5   = 000240           127#
PR6   = 000300           128#
PR7   = 000340           129#
PS    = 177776            86#     87
PSW   = 177776            87#   5912*
PWRVEC= 000024           194#    941*    942*   5778*   5779*   5788*   5794*   5806*   5807*
RESMON  031130           996    6005#
RESREG= 104407          5762#   5843    6233
RESVEC= 000010           189#
RSET  = 104410          2111    2197    2281    2368    2454    2542    2628    2727    2826    2925    3024    3123    3222
                        3321    3420    3548    3657    3769    3880    3991    4102    4213    4323    4381    4447    4512
                        4632    4772    4913    5055    5177    5298    5764#   5892    6006    6029    6049
R0    =%000000           101#    109     988*    992*   1021*   1023*   1027*   1031    1034*   1035    1052    1054    1060
```

```
                    1066    1072    1078    1084    1118*   1119    1127    1135    1169*   1171    1177    1180    1183
                    1211*   1215*   1216    1222*   1257*   1259    1270*   1272    1283*   1285    1353*   1356    1368*
                    1370    1381*   1383    1448*   1449*   1450    1454    1541*   1542*   1543    1547    1641*   1645*
                    1649*   1655    1656*   1659    1660*   1682*   1705*   1706    1715*   1718*   1719*   1720    1723*
                    1726    1842*   1846*   1847    1849    1853    1857    1863    1871    1873    1879    1962*   1972
                    3461*   3473*   3474*   3477*   3576*   3588*   3589*   3592*   3684*   3696*   3697*   3700*   3796*
                    3808*   3809*   3812*   3907*   3919*   3920*   3923*   4018*   4030*   4031*   4034*   4129*   4141*
                    4142*   4145*   4243*   4249*   4250*   4251*   4252*   4253*   4254*   4255*   4256*   4266*   4275
                    4408*   4409    4410    4416    4473*   4474    4475    4481    4664*   4676*   4677*   4680*   4804*
                    4816*   4817*   4820*   4945*   4957*   4958*   4961*   5089*   5104    5114    5125    5135    5172
                    5174*   5210*   5225    5235    5246    5256    5293    5295*   5328*   5332*   5335    5478    5503*
                    5527    5528*   5529    5532*   5668    5678*   5682    5698    5699    5712*   5731    5732*   5733
                    5734*   5735*   5736*   5737*   5780    5805*   5833*   5837    5847*   5849*   5851*   5869*   5870
                    5872*   5873    5990    6005*   6008*   6009    6061    6063*   6064*   6065*   6068*   6069*   6071*
                    6074    6075    6078*   6087    6088*   6089*   6095*   6096*   6097*   6098    6104    6111    6114*
                    6115    6120    6126    6136    6143    6149    6156    6162    6176*   6182*   6207*   6208*   6209*
                    6216*   6218
R1      =%000001     102#    110     989*    991*    1020*   1022*   1122*   1123    1128    1170*   1173    1178    1216*
                    1219    1226    1258*   1260*   1271*   1273*   1284*   1286*   1354*   1357*   1369*   1371*   1382*
                    1384*   1450*   1451*   1452    1460    1461    1485    1543*   1544*   1545    1553    1554    1578
                    1639*   1643    1650*   1661*   1686*   1707*   1708    1714*   1724*   1726    1847*   1848*   1902
                    1904    1911    1959*   1965*   1966    1967*   1972    1988    2147    2232    2317    2404    2494
                    2578    3469*   3470*   3471    3584*   3585*   3586    3609    3692*   3693*   3694    3719    3804*
                    3805*   3806    3915*   3916*   3917    4026*   4027*   4028    4137*   4138*   4139    4244*   4247
                    4362    4560    4584    4672*   4673*   4674    4698    4724    4812*   4813*   4814    4840    4865
                    4953*   4954*   4955    4981    5007    5103*   5104*   5106    5124*   5125*   5128    5224*   5225*
                    5227    5245*   5246*   5249    5479    5502*   5669    5682*   5683    5687    5711*   5781    5804*
                    5829*   5833    5834*   5846*   5848*   5850*   5988*   5990    5993*   6014*   6016    6062    6066*
                    6070*   6077*   6110    6111*   6117    6122    6128    6138    6145    6151    6164    6177    6181*
                    6200*   6201    6213*   6214*   6215*   6222
R10     =%000000     109#
R11     =%000001     110#
R12     =%000002     111#
R13     =%000003     112#
R14     =%000004     113#
R15     =%000005     114#
R2      =%000002     103#    111     990*    991     1212*   1215    1217    1221*   1225    1265*   1278*   1291*   1299
                    1304    1309    1362*   1363    1376*   1377    1389*   1390    1397    1402    1407    1454*   1455*
                    1456    1473    1474    1497    1547*   1548*   1549    1566    1567    1590    1640*   1645    1648*
                    1653*   1657*   1690*   1709    1716*   1721*   1725*   1729*   1849*   1850*   1902    1904    1913
                    1966*   1997    2002    2061*   2065*   2066*   2136*   2146*   2150*   2221*   2231*   2234*   2305*
                    2316*   2320*   2392*   2403*   2406*   2473*   2490*   2491    2492*   2576*   2580*   2656*   2675*
                    2676*   2755*   2774*   2775*   2854*   2873*   2874*   2953*   2972*   2973*   3052*   3071*   3072*
                    3151*   3170*   3171*   3250*   3269*   3270*   3349*   3368*   3369*   3463*   3468*   3478*   3484*
                    3486*   3488*   3578*   3583*   3593*   3598*   3608*   3610*   3686*   3691*   3701*   3708*   3718*
                    3720*   3798*   3803*   3813*   3820*   3830*   3831*   3909*   3914*   3924*   3931*   3941*   3942*
                    4020*   4025*   4035*   4042*   4052*   4053*   4131*   4136*   4146*   4153*   4163*   4164*   4245*
                    4247*   4351*   4360*   4368*   4426*   4428*   4429*   4491*   4493*   4494*   4550*   4559*   4561*
                    4574*   4583*   4585*   4666*   4671*   4681*   4688*   4697*   4699*   4714*   4723*   4725*   4806*
                    4811*   4821*   4830*   4839*   4841*   4855*   4864*   4866*   4947*   4952*   4962*   4971*   4980*
                    4982*   4997*   5006*   5008*   5093    5099*   5105*   5109*   5110*   5122*   5127*   5130*   5131*
                    5214    5220*   5226*   5230*   5231*   5243*   5248*   5251*   5252*   5480    5501*   5670    5681*
                    5685*   5688    5695*   5696*   5697    5702*   5710*   5782    5803*   5835*   5836    5837*   5852*
                    5853*   5989*   5992*   6015*   6016*   6219*   6220*   6223*   6225
R3      =%000003     104#    112     1217*   1218*   1219    1227    1470*   1471*   1472    1479    1494*   1495*   1496
                    1502    1563*   1564*   1565    1572    1587*   1588*   1589    1595    1694*   1710    1717*   1720*
```

```
                    3464*   3476*   3579*   3591*   3687*   3699*   3799*   3811*   3910*   3922*   4021*   4033*   4132*
                    4144*   4246*   4248*   4667*   4679*   4807*   4819*   4948*   4960*   5330*   5481    5500*   5610
                    5619*   5625*   5626*   5629*   5634*   5635*   5636    5645*   5671    5679*   5680*   5694*   5697*
                    5706*   5707*   5709*   5783    5802*   5832*   5845*   5852    6007*   6019    6218*   6222*   6226
R4      =%000004     105#    113     1698*   1711    1960*   1965    1992    1997    2001    2006*   2060*   2065    2071
                    2135*   2146    2153    2220*   2231    2237    2304*   2316    2324    2391*   2403    2410    2472*
                    2490    2498    2564*   2565*   2576    2584    2655*   2675    2683    2754*   2774    2782    2853*
                    2873    2881    2952*   2972    2980    3051*   3071    3079    3150*   3170    3178    3249*   3269
                    3277    3348*   3369    3376    3465*   3469    3473    3475*   3580*   3584    3588    3590*   3597*
                    3608    3613    3688*   3692    3696    3698*   3707*   3718    3725    3800*   3804    3808    3810*
                    3819*   3830    3836    3911*   3915    3919    3921*   3930*   3941    3947    4022*   4026    4030
                    4032*   4041*   4052    4058    4133*   4137    4141    4143*   4152*   4163    4169    4350*   4360
                    4425*   4428    4490*   4493    4549*   4559    4573*   4583    4668*   4672    4676    4678*   4687*
                    4697    4713*   4723    4808*   4812    4816    4818*   4829*   4839    4854*   4864    4949*   4953
                    4957    4959*   4970*   4980    4996*   5006    5098*   5105    5121*   5127    5219*   5226    5242*
                    5248    5482    5499*   5611    5613*   5614*   5615*   5616    5617*   5631    5633*   5641*   5644*
                    5784    5801*   5831*   5839*   6013*   6017*
R5      =%000005     106#    114     1663*   1682    1686    1690    1694    1698    1702*   1712    1906*   1907*   1908
                    1988*   1993    2475*   2491    2660*   2661    2662    2668    2759*   2760    2761    2767    2858*
                    2859    2860    2866    2957*   2958    2959    2965    3056*   3057    3058    3064    3155*   3156
                    3157    3163    3254*   3255    3256    3262    3353*   3354    3355    3361    3466*   3471*   3472*
                    3581*   3586*   3587*   3689*   3694*   3695*   3801*   3806*   3807*   3912*   3917*   3918*   4023*
                    4028*   4029*   4134*   4139*   4140*   4669*   4674*   4675*   4809*   4814*   4815*   4950*   4955*
                    4956*   5483    5498*   5612    5618*   5620*   5622*   5623*   5624*   5625    5643*   5672    5674*
                    5676*   5683*   5687*   5702    5708*   5785    5800*   5830*   5836*   5841*   5842
R6      =%000006     107#    929*    930*    931
R7      =%000007     108#
SAVREG= 104406      5761#   5828    6199
SDPAR0= 172260       319#
SDPAR1= 172262       320#
SDPAR2= 172264       321#
SDPAR3= 172266       322#
SDPAR4= 172270       323#
SDPAR5= 172272       324#
SDPAR6= 172274       325#
SDPAR7= 172276       326#
SDPDR0= 172220       297#
SDPDR1= 172222       298#
SDPDR2= 172224       299#
SDPDR3= 172226       300#
SDPDR4= 172230       301#
SDPDR5= 172232       302#
SDPDR6= 172234       303#
SDPDR7= 172236       304#
SIPAR0= 172240       308#
SIPAR1= 172242       309#
SIPAR2= 172244       310#
SIPAR3= 172246       311#
SIPAR4= 172250       312#
SIPAR5= 172252       313#
SIPAR6= 172254       314#
SIPAR7= 172256       315#
SIPDR0= 172200       286#
SIPDR1= 172202       287#
SIPDR2= 172204       288#
SIPDR3= 172206       289#
```

```
SIPDR4= 172210           290#
SIPDR5= 172212           291#
SIPDR6= 172214           292#
SIPDR7= 172216           293#
SIZE  = 104413          5767#
SIZEHI= 177762           218#
SIZELO= 177760           216#  6064
SKAD    030646          1016*  1108*  1158*  1204*  1246*  1343*  1440*  1533*  1624*  1835*  1943*  2050*  2126*
                        2211*  2295*  2382*  2468*  2556*  2646*  2745*  2844*  2943*  3042*  3141*  3240*  3339*
                        3441*  3665*  3674*  3786*  3897*  4008*  4119*  4232*  4339*  4399*  4464*  4637*  4654*
                        4794*  4935*  5083*  5204*  5893   5895*  6030   6050
SKBADR  030740          5769   5928#
SKBCNR  031002          5770   5944#
SKBERR  030764          5769   5937#
SKBHMR  031036          5772   5958#
SKBMNR  031020          5771   5951#
SKIPT = 104411          2672   2771   2870   2969   3068   3167   3266   3365   4420   4485   5765#  5865   5981
                        5965
SKPBAD= 104414          1163   5768#
SKPBCN= 104416          1113   1209   1251   1348   1445   1538   1629   1840   1947   2055   2131   2216   2300
                        2387   2561   2651   2750   2849   2948   3047   3146   3245   3344   3446   3570   3679
                        3791   3902   4013   4124   4238   4344   4404   4469   4542   4659   4799   4940   5770#
SKPBER= 104415          1164   2054   2130   2215   2299   2386   2560   2650   2749   2848   2947   3046   3145
                        3244   3343   3445   3569   3678   3790   3901   4012   4123   4237   4343   4403   4468
                        4541   4658   4798   4939   5769#
SKPBHM= 104420          1252   1349   1446   1539   1630   1841   2057   2133   2218   2302   2389   2563   2653
                        2752   2851   2950   3049   3148   3247   3346   3448   3572   3681   3793   3904   4015
                        4126   4240   4346   4406   4471   4544   4661   4801   4942   5772#
SKPBMN= 104417          1114   1948   2056   2132   2217   2301   2388   2562   2652   2751   2850   2949   3048
                        3147   3246   3345   3447   3571   3680   3792   3903   4014   4125   4239   4345   4405
                        4470   4543   4660   4800   4941   5771#
SKRNG   031052          5935   5942   5949   5956   5964#
SP    =%000006           115#   116    117    118    933*   951*   959*   963   1048   1050   1051   2023   2024
                        2079   2099   2100   2162   2185   2186   2246   2269   2270   2333   2356   2357   2419
                        2442   2443   2507   2530   2531   2593   2616   2617   2692   2715   2716   2791   2814
                        2815   2890   2913   2914   2989   3012   3013   3088   3111   3112   3187   3210   3211
                        3286   3309   3310   3385   3408   3409   3459*  3506   3515   3516   3517   3527*  3622
                        3645   3646   3734   3757   3758   3845   3868   3869   3956   3979   3980   4067   4090
                        4091   4178   4201   4202   4291   4292   4302   4373   4374   4438   4439   4503   4504
                        4596*  4736*  4877*  5019*  5158   5169   5279   5290   5325*  5367*  5370   5372   5373
                        5378*  5379*  5380   5403   5404   5408*  5434   5447*  5450*  5478*  5479*  5480*  5481*
                        5482*  5483*  5484*  5485*  5486*  5487*  5494*  5495*  5496*  5497*  5498   5499   5500
                        5501   5502   5503   5527*  5528   5529*  5531   5532   5533*  5535   5537   5539   5545
                        5547*  5549*  5557*  5561   5565   5566   5570   5602*  5603   5604   5605*  5610*  5611*
                        5612*  5618   5643   5644   5645   5646*  5647*  5668*  5669*  5670*  5671*  5672*  5673*
                        5674   5677*  5690   5692*  5694   5704   5706   5708   5709   5710   5711   5712   5714*
                        5715*  5731*  5732   5742*  5743*  5780*  5781*  5782*  5783*  5784*  5785*  5786*  5787
                        5795*  5799   5800   5801   5802   5803   5804   5805   5810*  5829   5842*  5860   5863
                        5878   5879   5882   5888   5890   5904   5905*  5964   6048   6061*  6062*  6063   6075*
                        6077   6078   6087*  6091*  6110*  6117*  6122*  6128*  6130*  6131   6138*  6145*  6164*
                        6166   6181   6182   6200   6227*  6229*  6230   6234*
SPUR    030522          1002   1019   1111   1443   1536   1627   1838   2017   2018   2031   4235   5086   5154
                        5165   5207   5275   5286   5868#  5874   5902
SR0   = 177572           235#
SR1   = 177574           236#
SR2   = 177576           237#
```

```
SR3    = 172516            239#
SSP    =%000006            117#
STACK  = 001100             80#     91      92      93     933    5905
START    003016            488     926#   5811
STKLMT= 177774             88#
STOP     031214           6021    6023#
SUPSTK= 000700             82#
SWR      001140            529#    931     953*    955     961*   5362   5376   5378   5386   5393   5430   5437   5442
                          5445    5786    5799*   6045
SWREG    000176            486#    961
SW0    = 000001            157#
SW00   = 000001            147#    157
SW01   = 000002            146#    156
SW02   = 000004            145#    155
SW03   = 000010            144#    154
SW04   = 000020            143#    153
SW05   = 000040            142#    152
SW06   = 000100            141#    151
SW07   = 000200            140#    150
SW08   = 000400            139#    149
SW09   = 001000            138#    148
SW1    = 000002            156#
SW10   = 002000            137#
SW11   = 004000            136#
SW12   = 010000            135#
SW13   = 020000            134#
SW14   = 040000            133#
SW15   = 100000            132#
SW2    = 000004            155#
SW3    = 000010            154#
SW4    = 000020            153#
SW5    = 000040            152#
SW6    = 000100            151#
SW7    = 000200            150#   6045
SW8    = 000400            149#
SW9    = 001000            148#
SYSTID= 177764             220#
SOMOM1= 000034             472#   1282
SOM1   = 000030            470#   1380    1470    1552    1560   1587   1642   1744   1870   2657   2756   3053   3152
                          3818    4040    4407    4472
S1MO   = 000044            469#   1367    1459    1467    1494   1563   1644   1741   1852   2855   2954   3251   3350
                          3929    4151
S1MOM1= 000054             471#   1269
TAB    = 000011            468#   6257    6262    6269    6283   6286   6292   6302   6311   6320   6323   6449   6454
                          6516    6521    6527    6537    6543   6548   7387   7511   7519   7525   7531   7536   7555
                          7560    7568    7577    7584    7590   7596   7599   7622   7627   7636   7644   7653   7660
                          7670    7679    7685    7691    7695   7702   7708   7712   7726   7738
TBITVE= 000014             190#
TESTR1= 140000             475#   1451    1544    1639    1848
TESTR2= 142000             476#   1455    1548    1640    1850
TESTR3= 144000             477#
TKVEC  = 000060            197#    986     996*    997*   6025*
TOP      003420            984     993#
TPVEC  = 000064            198#
TRAPVE= 000034             196#    939*    940*
TRTVEC= 000014             191#
```

TST1	003470	1012#											
TST10	005712	1440	1529#										
TST11	006242	1533	1620#										
TST12	007054	1624	1808#										
TST13	007106	1831#											
TST14	007342	1835	1939#										
TST15	007624	1943	2046#										
TST16	010122	2050	2122#										
TST17	010440	2126	2207#										
TST2	004044	1016	1105#										
TST20	010754	2211	2291#										
TST21	011274	2295	2378#										
TST22	011614	2382	2464#										
TST23	012130	2468	2552#										
TST24	012450	2555	2642#										
TST25	013014	2646	2741#										
TST26	013360	2745	2840#										
TST27	013724	2844	2939#										
TST3	004200	1108	1154#										
TST30	014270	2943	3038#										
TST31	014634	3042	3137#										
TST32	015200	3141	3236#										
TST33	015544	3240	3335#										
TST34	016110	3339	3437#										
TST35	016540	3441	3561#										
TST36	017160	3565	3670#										
TST37	017604	3674	3782#										
TST4	004342	1158	1199#										
TST40	020224	3786	3893#										
TST41	020644	3897	4004#										
TST42	021264	4008	4115#										
TST43	021704	4119	4228#										
TST44	022334	4232	4335#										
TST45	022510	4339	4395#										
TST46	022710	4399	4460#										
TST47	023110	4464	4533#										
TST5	004460	1204	1242#										
TST50	023474	4537	4650#										
TST51	024164	4654	4790#										
TST52	024664	4794	4931#										
TST53	025370	4935	5079#										
TST54	025744	5083	5200#										
TST55	026320	5204	5300#										
TST6	005012	1246	1339#										
TST7	005362	1343	1436#										
TVADFL	031720	6146*	6158*	6185#	6203								
TVADHI	031724	6192#	6202*	6225*									
TVADLO	031722	6191#	6201*	6226*	6227								
TYPDS =	104405	5326	5758#	6123									
TYPE =	104401	971	5324	5327	5432	5440	5540	5637	5713	5754#	5808	5933	5940	5947
		5954	5961	6011	6023	6085	6100	6102	6106	6108	6132	6152	6167	6173
		6231												
TYPOC =	104402	5755#	6092	6118										
TYPON =	104404	5757#												
TYPOS =	104403	5756#	6139											
TYPVAD	031726	6147	6159	6199#										

DEKBCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
UA     = 000054          5081#
UAER1    025630          5097    5151#
UAER2    025670          5119    5162#
UATMP1   025624          5103    5115    5148#   5155
UATMP2   025626          5124    5136    5149#   5166
UA1      025432          5091#   5175
UA2      025456          5094    5097#
UA3      025512          5112#
UA4      025532          5119#   5120    5160
UA5      025576          5133#
UA6      025616          5140#
UA7      025724          5095    5140    5172#
UA8      025742          5173    5177#
UB     = 000055          5202#
UBER1    026204          5218    5272#
UBER2    026244          5240    5283#
UBTMP1   026200          5224    5236    5269#   5276
UBTMP2   026202          5245    5257    5270#   5297
UB1      026006          5212#   5296
UB2      026032          5215    5218#
UB3      026066          5233#
UB4      026106          5240#   5241    5281
UB5      026152          5254#
UB6      026172          5261#
UB7      026300          5216    5261    5293#
UB8      026316          5294    5298#
UDPAR0=  177660           275#
UDPAR1=  177662           276#
UDPAR2=  177664           277#
UDPAR3=  177666           278#
UDPAR4=  177670           279#
UDPAR5=  177672           280#
UDPAR6=  177674           281#
UDPAR7=  177676           282#
UDPDR0=  177620           253#
UDPDR1=  177622           254#
UDPDR2=  177624           255#
UDPDR3=  177626           256#
UDPDR4=  177630           257#
UDPDR5=  177632           258#
UDPDR6=  177634           259#
UDPDR7=  177636           260#
UIPAR0=  177640           264#
UIPAR1=  177642           265#
UIPAR2=  177644           266#
UIPAR3=  177646           267#
UIPAR4=  177650           268#
UIPAR5=  177652           269#
UIPAR6=  177654           270#
UIPAR7=  177656           271#
UIPDR0=  177600           242#
UIPDR1=  177602           243#
UIPDR2=  177604           244#
UIPDR3=  177606           245#
UIPDR4=  177610           246#
UIPDR5=  177612           247#
```

DEKBC8.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
UIPDR6= 177614            248#
UIPDR7= 177616            249#
USESTK= 000600             83#
USP   =%000006            118#
$AUTOB  001134            525#
$BDADR  001122            520#
$BDDAT  001126            522#
$BELL   001306            582#  5432  5459
$CHARC  027426           5542*  5552*  5559  5568*  5573#
$CKSWR= ****** U          5761
$CMTAG  001100             509#   928   929   937   943   944   945
$CM1  = 000024             540#   541#  542#  543#  544#  545#  546#  547#  548#  549#  550#  551#  552#
                           553#   554#  555#  556#  557#  558#  559#  560#
$CM2  = 000050             540#   541#  542#  543#  544#  545#  546#  547#  548#  549#  550#  551#  552#
                           553#   554#  555#  556#  557#  558#  559#  560#
$CM3  = 000024             538#   540
$CM4  = 000024             560#   561#  562#  563#  564#  565#  566#  567#  568#  569#  570#  571#  572#
                           573#   574#  575#  576#  577#  578#  579#  580#
$CRLF   001313             584#  5440  5459  5541  5576  6086  6103  6109
$DBLK   030074            5679   5713  5721#
$DB20   030354            5828#  6129  6165  6228
$DOAGN  026434            5320   5329  5333  5339#
$DTBL   030064            5682   5717#
$ENDAD  026424             496    969  5335#  5452
$ENDCT  026354             943   5322#
$ENDMG  026443            5324   5343#
$ENULL  026440            5327   5342#
$EOP    026320            5312#
$EOPCT  026346             943*  5319#  5323
$ERFLG  001103             511#  5352   5382  5384  5390*  5411  5427*  5459
$ERMAX  001115             517#   946*  5384  5406*  5411
$ERROR  026736             937   5426#
$ERRPC  001116             518#  5434*  5435*  5436  5459  6091  7410  7862  7865  7871  7884  7887  7890
                          7893   7908   7911  7914  7922  7936  7939  7944  7948  7951   ^37  7960  7969
                          7973
$ERRTB  001316             601#  6097
$ERTTL  001112             515#  5433*  5459
$ESCAP  001304             581#   945*  5405*  5448  5450  5459
$FILLC  001156             536#  5545   5576
$FILLS  001155             535#  5576
$GDADR  001120             519#
$GDDAT  001124             521#
$GET42  026376            5328#
$GTSWR= ****** U          5760
$HD   = 000003              59    60
$ICNT   001104             512#  5397*  5398  5400*  5410
$ILLUP  030346            5778   5794  5813#
$INTAG  001135             526#
$ITEMB  001114             516#  5436*  5459  5889*  6089
$LF     001314             585#  5459   5576
$LPADR  001106             513#   947*  5388*  5403*  5408  5410
$LPERR  001110             514#   948*  1028*  1115*  1166*  1213*  1255*  1268*  1281*  1352*  1366*  1379*  1633*
                          1961*  5091*  5120*  5212*  5241*  5388  5404*  5410  5447
$MAIL = ****** U           965   5403  5442  5529
$MXCNT  026734            5401   5410#
$NULL   001154             534#  5547  5576
```

```
$NWTST= 000001          1005#   1007    1097#   1099    1141#   1143    1190#   1192    1234#   1236    1326#   1328    1422#
                        1424    1515#   1517    1608#   1610    1791#   1793    1818#   1820    1924#   1926    2036#   2038
                        2114#   2116    2199#   2201    2283#   2285    2370#   2372    2456#   2458    2544#   2546    2631#
                        2633    2730#   2732    2829#   2831    2928#   2930    3027#   3029    3126#   3128    3225#   3227
                        3324#   3326    3426#   3428    3550#   3552    3659#   3661    3771#   3773    3882#   3884    3993#
                        3995    4104#   4106    4215#   4217    4325#   4327    4385#   4387    4450#   4452    4518#   4520
                        4635#   4637    4775#   4777    4916#   4918    5058#   5060    5179#   5181
$OCNT   027654          5609*   5638*   5651#
$OCTVL  030456          5830    5855#
$OMODE  027656          5604*   5608*   5613    5616*   5627*   5653#
$OVER   026720          5363    5381    5389    5399    5407#
$PASS   001100           509#   5316*   5317*   5325    5342    5395    5411
$PWRAD  030342          5811#
$PWRDN  030202           941    5778#   5806
$PWRMG  030336          5809#
$PWRUP  030254          5788    5794#
$QUES   001312           583#   5459    5576
$RDCHR= ****** U        5761
$RDDEC= ****** U        5761
$RDLIN= ****** U        5761
$RDOCT= ****** U        5761
$REGAD  001160           538#
$REG0   001162           540#
$REG1   001164           541#
$REG10  001202           548#
$REG11  001204           549#
$REG12  001206           550#
$REG13  001210           551#
$REG14  001212           552#
$REG15  001214           553#
$REG16  001216           554#
$REG17  001220           555#
$REG2   001166           542#
$REG20  001222           556#
$REG21  001224           557#
$REG22  001226           558#
$REG23  001230           559#
$REG3   001170           543#
$REG4   001172           544#
$REG5   001174           545#
$REG6   001176           546#
$REG7   001200           547#
$RESRE  027154          5493#   5762
$RTNAD  026436          5341#
$R2A  = ****** U        5763
$SAVRE  027116          5477#   5761
$SAVR6  030352          5787*   5795    5796*   5797*   5815#
$SCOPE  026460           935    5361#
$SETUP= 000037           462#    934     935     937     939     941     943     944     945     947     969     972    5314
                        5362    5427    5445    5452
$STUP = 177777           462#
$SVLAD  026672          5371    5402#
$SVPC = 000204           494#    499
$SWR  = 167400            59     60#     61#     68      69      70      71      72      73      74     580     581     582
                         944     945     947     948    1013    1106    1155    1200    1243    1340    1437    1530    1621
                        1809    1832    1940    2047    2123    2208    2292    2379    2465    2553    2643    2742    2841
```

```
                    2940    3039    3138    3237    3336    3438    3562    3671    3783    3894    4005    4116    4229
                    4336    4396    4461    4534    4651    4791    4932    5080    5201    5309    5315    5330    5340
                    5342    5353    5354    5355    5356    5357    5362    5374    5376    5377    5382    5383    5384
                    5391    5392    5393    5404    5407    5410    5418    5419    5420    5421    5422    5430    5437
                    5442    5445    5459    5812
$$WRMK= 000200        62*     74      75    5357    5358    5378    5379
$TAB    032174      6174    6257*
$TIMES  001302       590*    944*   1013*   1155*   1200*   1243*   1340*   1437*   1530*   1621*   1832*   1940*   2047*
                    2123*   2208*   2292*   2379*   2465*   2553*   2643*   2742*   2841*   2940*   3039*   3138*   3237*
                    3336*   3438*   3562*   3671*   3783*   3894*   4005*   4116*   4229*   4336*   4396*   4461*   4534*
                    4651*   4791*   4932*   5080*   5201*   5315*   5391*   5398    5401*   5410
$TKB    001146       531*    998*   6005    6027*
$TKS    001144       530*    999*   6028*
$TMP0   001232       560*   1018*   1110*   1160*   1206*   1248*   1345*   1442*   1535*   1626*   1837*   1945*   2052*
                    2129*   2213*   2297*   2384*   2470*   2558*   2648*   2747*   2846*   2945*   3044*   3143*   3242*
                    3341*   3443*   3567*   3676*   3788*   3899*   4010*   4121*   4234*   4341*   4401*   4466*   4539*
                    4656*   4796*   4937*   5085*   5206*   7410    7862    7865    7868    7871    7884    7887    7890
                    7893    7908    7911    7914    7922    7936    7939    7944    7948    7951    7957    7960    7969
                    7973    7978
$TMP1   001234       561*   1050*   1176*   1864*   1880*   2021*   2669*   2768*   2867*   2966*   3065*   3164*   3263*
                    3362*   4417*   4482*   5860*   5876*   5888*   7862    7865    7868    7871    7944    7978
$TMP10  001252       568*   1452*   1545*   7908    7914
$TMP11  001254       569*   1453*   1546*
$TMP12  001256       570*   1456*   1549*   7911    7917
$TMP13  001260       571*   1457*   1550*
$TMP14  001262       572*
$TMP15  001264       573*
$TMP16  001266       574*
$TMP17  001270       575*
$TMP2   001236       562*   1127*   1135*   1177*   1225*   1299*   1304*   1309*   1397*   1402*   1407*   1466*   1490*
                    1501*   1559*   1583*   1594*   1783*   1863*   1879*   1911*   1992*   2001*   2022*   2071*   2085*
                    2099*   2153*   2170*   2185*   2237*   2254*   2269*   2324*   2341*   2356*   2410*   2427*   2442*
                    2498*   2515*   2530*   2584*   2601*   2616*   2668*   2683*   2700*   2715*   2767*   2782*   2799*
                    2814*   2866*   2881*   2898*   2913*   2965*   2980*   2997*   3012*   3064*   3079*   3096*   3111*
                    3163*   3178*   3195*   3210*   3262*   3277*   3294*   3309*   3361*   3376*   3393*   3408*   3501*
                    3516*   3536*   3613*   3630*   3645*   3725*   3742*   3757*   3836*   3853*   3868*   3947*   3964*
                    3979*   4058*   4075*   4090*   4169*   4186*   4201*   4278*   4291*   4311*   4373*   4416*   4438*
                    4481*   4503*   4564*   4588*   4603*   4619*   4702*   4728*   4743*   4759*   4844*   4869*   4894*
                    4900*   4985*   5011*   5026*   5042*   5114*   5135*   5235*   5256*   5861*   5877*   7410    7862
                    7884    7887    7890    7893    7908    7911    7914    7922    7936    7939    7948    7951    7957
                    7960    7969    7973    7978
$TMP20  001272       576*
$TMP21  001274       577*
$TMP22  001276       578*
$TMP23  001300       579*
$TMP3   001240       563*   1052*   1128*   1178*   1226*   1467*   1479*   1491*   1502*   1560*   1572*   1584*   1595*
                    1912*   1993*   2002*   2023*   2086*   2101*   2171*   2187*   2255*   2271*   2342*   2358*   2428*
                    2444*   2516*   2532*   2602*   2618*   2701*   2717*   2800*   2816*   2899*   2915*   2998*   3014*
                    3097*   3113*   3196*   3212*   3295*   3311*   3394*   3410*   3518*   3537*   3631*   3647*   3743*
                    3759*   3854*   3870*   3965*   3981*   4076*   4092*   4187*   4203*   4279*   4293*   4312*   4375*
                    4440*   4505*   4604*   4620*   4744*   4760*   4885*   4901*   5027*   5043*   5115*   5136*   5236*
                    5257*   5862*   5878*   7410    7865    7871    7884    7890    7893    7908    7911    7917    7939
                    7944    7951    7960    7969    7973    7978
$TMP4   001242       564*   1053*   1227*   1913*   2020*   2087*   2102*   2172*   2188*   2256*   2272*   2343*   2359*
                    2429*   2445*   2517*   2533*   2603*   2619*   2702*   2718*   2801*   2817*   2900*   2916*   2999*
                    3015*   3098*   3114*   3197*   3213*   3296*   3312*   3395*   3411*   3519*   3538*   3632*   3648*
```

```
                    3744*   3760*   3855*   3871*   3966*   3982*   4077*   4093*   4188*   4204*   4294*   4313*   4376*
                    4441*   4506*   4605*   4621*   4745*   4761*   4886*   4902*   5028*   5044*   5116*   5137*   5237*
                    5258*   5875*   7865    7893    7936    7944    7957    7969    7975
$TMP5   001244       565*   1914*   2103*   2189*   2273*   2360*   2446*   2534*   2620*   2719*   2818*   2917*   3016*
                    3115*   3214*   3313*   3412*   3520*   3649*   3761*   3872*   3983*   4094*   4205*   4295*   4377*
                    4442*   4507*   4606*   4746*   4887*   5029*   7951    7960
$TMP6   001246       566*   2104*   2190*   2274*   2361*   2447*   2535*   2621*   2720*   2819*   2918*   3017*   3116*
                    3215*   3314*   3413*   3521*   3650*   3762*   3873*   3984*   4095*   4206*   4296*   4379*   4443*
                    4508*   4607*   4747*   4888*   5030*   7944    7962    7969    7975
$TMP7   001250       567*   2105*   2191*   2275*   2362*   2448*   2536*   2622*   2721*   2820*   2919*   3018*   3117*
                    3216*   3315*   3414*   3522*   3651*   3763*   3874*   3985*   4096*   4207*   4297*   4608*   4748*
                    4889*   5031*   7951    7962
$TN     = 000055     59*   1005    1013*   1014    1015    1097    1106*   1107    1141    1155*   1156    1157    1190
                    1200*   1202    1203    1234    1243*   1244    1245    1326    1340*   1341    1342    1422    1437*
                    1438    1439    1515    1530*   1531    1532    1608    1621*   1622    1623    1791    1809*   1818
                    1832*   1833    1834    1924    1940*   1941    1942    2036    2047*   2048    2049    2114    2123*
                    2124    2125    2199    2208*   2209    2210    2283    2292*   2293    2294    2370    2379*   2380
                    2081    2456    2465*   2466    2467    2544    2553*   2554    2555    2631    2643*   2644    2645
                    2730    2742*   2743    2744    2829    2841*   2842    2843    2928    2940*   2941    2942    3027
                    3039*   3040    3041    3126    3138*   3139    3140    3225    3237*   3238    3239    3324    3336*
                    3337    3338    3426    3438*   3439    3440    3550    3562*   3563    3564    3659    3671*   3672
                    3673    3771    3783*   3784    3785    3882    3894*   3895    3896    3993    4005*   4006    4007
                    4104    4116*   4117    4118    4215    4229*   4230    4231    4325    4336*   4337    4338    4385
                    4396*   4397    4398    4450    4461*   4462    4463    4518    4534*   4535    4536    4635    4651*
                    4652    4653    4775    4791*   4792    4793    4916    4932*   4933    4934    5058    5080*   5091
                    5082    5179    5201*   5202    5203
$TPB    001152       533*   5565*   5576
$TPFLG  001157       537*   5523    5576
$TPS    001150       532*   5563    5576
$TRAP   030104       939    5731*
$TRAP2  030126      5742*   5753
$TRP    = 000021    5746*   5755*   5756*   5757*   5758*   5759*   5761    5762*   5763*   5764    5765*   5766*   5767*
                    5768*   5769*   5770*   5771*   5772*   5773*
$TRPAD  030140      5735    5753*
$TSTNM  001102       510*    926*   1018    1110    1160    1206    1248    1345    1442    1535    1626    1837    1945
                    2052    2128    2213    2297    2384    2470    2558    2648    2747    2846    2945    3044    3143
                    3242    3341    3443    3567    3676    3788    3899    4010    4121    4234    4341    4401    4466
                    4539    4656    4796    4937    5085    5206    5314*   5352    5380    5402*   5407    5411    5429
                    5459
$TYPBN= ****** U    5759
$TYPDS  027660      5667*   5758
$TYPE   027212      5523*   5746    5754
$TYPEC  027362      5544    5551    5558    5563*   5564
$TYPEX  027430      5569    5571    5574*
$TYPOC  027456      5607*   5755
$TYPON  027472      5606    5609*   5757
$TYPOS  027432      5602*   5756
$XTSTR  026470      5365*
$$GET4= 000001      5330*   5332*
$OFILL  027655      5603*   5607*   5617    5652*
$40CAT= ****** U    5362    5439
.       = 057312     480*    484*    494     495*    497*    499*    507*    586     932     947     948    1672    1675*
                    1977    2140    2143*   2225    2228*   2309    2312*   2396    2399*   2484    2487*   2570    2573*
                    3602    3605*   3711    3714*   3823    3826*   3934    3937*   4045    4048*   4156    4159*   4354
                    4357*   4553    4556*   4577    4580*   4691    4694*   4717    4720*   4833    4836*   4858    4861*
                    4974    4977*   5000    5003*   5143    5146*   5264    5267*   5342    5346    5410    5411    5459
```

5575 5721# 5790 5814 5855# 7409# 7859# 7982#

```
ADDDPA      37#
ADJUST      23#  1671  2139  2224  2308  2395  2483  2569  3601  3710  3822  3933  4044  4155  4363
          4552  4576  4690  4716  4832  4857  4973  4999  5142  5263
BYTLT1      26#
BYTLT2      25#
CHAIN        6#  5330
CLRMAC      28#  5900
CLRRFL      22#  1020
CMPDPA      36#
CNVDPA      35#
CNVMAP      30#
COMMEN       1#   461#
ENDCOM       1#   461#
ERCLR        6#  5456
ERROR       84#  1057  1063  1069  1075  1081  1087  1129  1136  1179  1228  1300  1305  1310  1398
          1403  1408  1468  1480  1492  1503  1561  1573  1585  1596  1784  1865  1881  1916  1994
          2003  2025  2072  2088  2106  2155  2174  2193  2239  2258  2277  2326  2345  2364  2412
          2431  2450  2500  2519  2538  2586  2605  2624  2670  2685  2704  2723  2769  2784  2803
          2822  2868  2883  2902  2921  2967  2982  3001  3020  3066  3081  3100  3119  3165  3180
          3199  3218  3264  3279  3298  3317  3363  3378  3397  3416  3502  3509  3523  3539  3615
          3634  3653  3727  3746  3765  3838  3857  3876  3949  3968  3987  4060  4079  4098  4171
          4190  4209  4280  4298  4314  4379  4418  4445  4483  4510  4565  4589  4610  4623  4703
          4729  4750  4763  4845  4870  4891  4904  4986  5012  5033  5046  5117  5138  5238  5259
          5864  5880
ESCAPE       1#   461#
GETPRI       1#   461#
GETSWR       1#   461#   972#
IIMAC1      12#
ITEMAC      24#   606
JJM1        43#
KKM1        33#
KKM2        34#
KMAC1       24#  1422  1515
KMAC5       17#  4398  4463
MMAC1       12#  2158  2242  2329  2415  2503  2589  2688  2787  2886  2985  3084  3183  3282  3381
          3618  3730  3841  3952  4063  4174
MMAC2       13#  2151  2235  2322  2408  2496  2582  2681  2780  2879  2978  3077  3176  3275  3374
          3611  3723  3834  3945  4056  4167
MMAC3       14#  2645  2744  2843  2942  3041  3140  3239  3338
MMAC4       15#  3785  3896  4007  4118
MMAC5       16#  3673
MMM1        40#
MMM2        41#
MSG       1004#  1007  1097#  1099  1189#  1192  1233#  1236  1326#  1328  1422#  1424  1515#  1517  1608#
          1610  1818#  1820  1924#  1926  2035#  2038  2114#  2116  2199#  2201  2283#  2285  2370#  2372
          2456#  2458  2544#  2546  2631#  2633  2730#  2732  2829#  2831  2928#  2930  3027#  3029  3126#
          3128  3225#  3227  3324#  3326  3426#  3428  3550#  3552  3659#  3661  3771#  3773  3882#  3884
          3993#  3995  4104#  4106  4215#  4217  4325#  4327  4385#  4387  4450#  4452  4518#  4520  4635#
          4637  4775#  4777  4916#  4918  5058#  5060  5179#  5181
MSG101    4516#  4528  4530  4645  4787
MSG102    4517#  4647  4785  4926  4928
MSG20     1421#  1424  1517
MSG200    5058#  5060  5181
MSG25     2033#  2116  2201  2285  2372  2458  2546
MSG27     2630#  2633  2732  2831  2930  3029  3128  3227  3326
MSG30     1791#  1793
```

DEKBCB.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
MSG300   7218*   7260    7302    7345
MSG31    3423*   3428    3552    3661    3773    3884    3995    4106
MSG32    3424*   3552    3661
MSG33    3425*   3773    3884    3995    4106
MSG6     1141*   1143
MSG98    4383*   4387    4452
MSG99    4515*   4520    4637    4777    4918
MULT        1*    461*
NEWTST      1*    461*   1005    1097    1141    1190    1234    1326    1422    1515    1608    1791    1818    1924    2036
         2114    2199    2283    2370    2456    2544    2631    2730    2829    2928    3027    3126    3225    3324    3426
         3550    3659    3771    3882    3993    4104    4215    4325    4385    4450    4518    4635    4775    4916    5058
         5179
NMAC1       18*  4547    4570    4685    4852
NMAC2       19*  4708    4825    4966    4991
NMAC3       21*  4594    4734    4875    5017
POP         1*    461*   5498    5708    5799    5800
PUSH        1*    461*   5478    5667    5780    5786
QQM1       42*
REPORT      1*    461*
SCOPE      85*   1012    1105    1154    1199    1242    1339    1436    1529    1620    1808    1831    1939    2046    2122
         2207    2291    2378    2464    2552    2642    2741    2840    2939    3038    3137    3236    3335    3437    3561
         3670    3782    3893    4004    4115    4228    4335    4395    4460    4533    4650    4790    4931    5079    5200
         5313
SETMAP     29*   3460    3575    3683    3795    3906    4017    4128    4663    4803    4944
SETMM      32*   4242
SETPRI      1*    461*
SETTRA   5746*   5755    5756    5757    5758    5761    5762    5764    5765    5766    5767    5768    5769    5770    5771
         5772
SETUP       1*    461*    927
SKBREX     22*   1112    1162    1208    1250    1347    1444    1537    1628    1839    1946    2053    2129    2214    2298
         2385    2559    2649    2748    2847    2946    3045    3144    3243    3342    3444    3568    3677    3789    3900
         4011    4122    4236    4342    4402    4467    4540    4657    4797    4938    5087    5208
SKIP        1*    461*
SLASH       1*    461*
SPACE     461*
SSKAD      37*   1015    1107    1157    1203    1245    1342    1439    1532    1623    1834    1942    2049    2125    2210
         2294    2381    2467    2555    2645    2744    2843    2942    3041    3140    3239    3338    3440    3564    3673
         3785    3896    4007    4118    4231    4338    4398    4463    4536    4653    4793    4934    5082    5203
STARS       1*    461*    492     503     586    1005    1011    1097    1104    1141    1153    1190    1198    1234    1241
         1326    1338    1422    1435    1515    1528    1608    1619    1791    1807    1818    1830    1924    1938    2036
         2045    2114    2121    2199    2206    2283    2290    2370    2377    2456    2463    2544    2551    2631    2641
         2730    2740    2829    2839    2928    2938    3027    3037    3126    3136    3225    3235    3324    3334    3426
         3436    3550    3560    3659    3669    3771    3781    3882    3892    3993    4003    4104    4114    4215    4227
         4325    4334    4385    4394    4450    4459    4518    4532    4635    4649    4775    4789    4916    4930    5058
         5079    5179    5199    5305    5349    5414    5462    5508    5579    5657    5725    5776    5792    5819
SWRSU       1*    461*    949*
THIT       38*   1858    1874    2663    2762    2861    2960    3059    3158    3257    3356    4411    4476
TOMAPO     27*
TOPAR3     31*
TRMTRP   5746*
TSTHIT     39*   1856    1872
TYPBIN      1*    461*
TYPDEC      1*    461*   5325
TYPNAM      1*    461*    965
TYPNUM      1*    461*
TYPOCS      1*    461*
```

```
TYPOCT      1#    461#
TYPTXT      1#    461#
UMAC1       7#   5058    5179
UMAC2       9#   5151    5162    5272    5283
UMAC3      10#   5112    5133    5233    5254
$$CMRE    501#    540     541     542     543     544     545     546     547     548     549     550     551     552     553
           554     555     556     557     558     559
$$CMTM    501#    560     561     562     563     564     565     566     567     568     569     570     571     572     573
           574     575     576     577     579     579
$$ESCA      1#    461#
$$NEWT      1#    461#   1005    1097    1141    1190    1234    1326    1422    1515    1608    1791    1818    1924    2036
          2114    2199    2283    2370    2456    2544    2631    2730    2829    2928    3027    3126    3225    3324    3426
          3550    3659    3771    3882    3993    4104    4215    4325    4385    4450    4518    4635    4775    4916    5058
          5179
$$SET     5746#   5755    5756    5757    5758    5761    5762    5764    5765    5766    5767    5768    5769    5770    5771
          5772
$$SKIP      1#    461#
.EQUAT      1#
.HEADE      1#     46#     49
.KT11       1#
.SETUP      1#     46#    462
.SWRHI      1#     46#     64
.SWRLO     75#
.$ACT1      1#     46#    490
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#     46#    478
.$CMTA      1#     46#    501
.$D82D      1#
.$DB20      1#     48#   5817
.$DIV       1#
.$EOP       1#     47#   5303
.$ERRO      1#     47#   5412
.$ERRT      1#
.$MULT      1#
.$POWE      1#     48#   5774
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#     47#   5460
.$SB2D      1#
.$SB20      1#
.$SCOP      1#     47#   5347
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#     48#   5723
.$TYPB      1#
.$TYPD      1#     48#   5655
.$TYPE      1#     47#   5506
.$TYPO      1#     47#   5577
.$40CA      1#
.1170       1#     46#     77
```

```
ADC     6223
ADD     1034  1451  1455  1544  1548  1656  1660  1731  1848  1850  2006  3474  3475  3527  3589
        3590  3697  3698  3809  3810  3920  3921  4031  4032  4142  4143  4596  4677  4678  4736
        4817  4818  4877  4958  4959  5019  5174  5295  5533  5605  5615  5687  5853  6097  6130
        6216  6222  6229
ASH     3470  3585  3693  3805  3916  4027  4138  4673  4813  4954  6096
ASHC    1718  6208  6220
ASL     1653  1657  5735  5993  6068  6209
ASLB    5692
ASR     5845
BCC     1654  1658  5693  5994
BEQ      970  1120  1124  1174  1181  1184  1321  1417  1499  1592  1727  1733  1740  1811  1814
        1909  1977  1989  1998  2082  2167  2251  2338  2424  2512  2598  2697  2796  2895  2994
        3093  3192  3291  3390  3508  3512  3533  3627  3739  3850  3961  4072  4183  4288  4308
        4600  4616  4740  4756  4881  4897  5023  5039  5153  5164  5173  5274  5285  5294  5329
        5333  5377  5381  5383  5387  5396  5428  5431  5446  5449  5536  5571  5632  5840  5871
        5991  6099  6105  6178  6204  6206
BGE     5399
BGT     5320  5639  5701  5839
BHI     5385
BIC     1218  1449  1471  1495  1542  1564  1588  1705  1707  1719  1846  5317  5379  5629  5852
        6008
BIS     1301  1306  1311  1399  1404  1409  4348  4422  4487  5634  5635  5695  5696  6071
BISB     999  6028
BIT     1044  1416  1462  1475  1486  1498  1555  1568  1579  1591  1860  1876  1908  1972  2014
        2665  2764  2863  2962  3061  3160  3259  3358  4283  4413  4478  5093  5214  5362  5376
        5386  5393  5430  5437  5445  5870  5990  6045  6205
BITB    5559
BLOS    1036
BLT     5550  5640  5684  5700
BMI     5691
BNE      932   956   968   984  1045  1049  1055  1061  1067  1073  1079  1085  1172  1220  1266
        1279  1292  1315  1364  1378  1391  1412  1463  1476  1487  1556  1569  1580  1861  1877
        2007  2015  2077  2093  2095  2160  2179  2181  2244  2263  2265  2331  2350  2352  2417
        2436  2438  2505  2524  2526  2591  2610  2612  2666  2690  2709  2711  2765  2789  2808
        2810  2864  2888  2907  2909  2963  2987  3006  3008  3062  3086  3105  3107  3161  3185
        3204  3206  3260  3284  3303  3305  3359  3383  3402  3404  3544  3546  3620  3639  3641
        3732  3751  3753  3843  3862  3864  3954  3973  3975  4065  4084  4086  4176  4195  4197
        4284  4319  4321  4414  4479  4598  4628  4630  4738  4768  4770  4879  4909  4911  5021
        5051  5053  5094  5156  5167  5215  5277  5288  5363  5394  5438  5453  5530  5538  5546
        5560  5567  5630  5689  5798  5929  5931  5938  5945  5952  5959  6010  6021  6046  6090
        6112  6116  6121  6127  6137  6144  6150  6157  6163
BPL     1737  5443  5524  5564  5628  5675  5705
BR       958   972  1058  1064  1070  1076  1082  1088  1132  1223  1230  1302  1307  1400  1405
        1414  1493  1507  1586  1600  1652  1664  1679  1683  1687  1691  1695  1699  1917  2008
        2028  2074  2090  2096  2109  2138  2157  2176  2182  2196  2223  2241  2260  2266  2280
        2307  2328  2347  2353  2367  2394  2414  2433  2439  2453  2482  2502  2521  2527  2541
        2568  2588  2607  2613  2627  2687  2706  2712  2726  2786  2805  2811  2825  2885  2904
        2910  2924  2984  3003  3009  3023  3083  3102  3108  3122  3182  3201  3207  3221  3281
        3300  3306  3320  3380  3399  3405  3419  3504  3525  3541  3600  3617  3636  3642  3656
        3709  3729  3748  3754  3768  3821  3840  3859  3865  3879  3932  3951  3970  3976  3990
        4043  4062  4081  4087  4101  4154  4173  4192  4198  4212  4281  4300  4316  4352  4370
        4431  4496  4551  4567  4575  4591  4625  4689  4705  4715  4731  4765  4831  4847  4856
        4872  4906  4972  4988  4998  5014  5048  5140  5261  5365  5371  5374  5389  5392  5526
        5543  5553  5562  5569  5606  5621  5642  5686  5703  5790  5814  5854  5935  5942  5949
        5956  6119  6124  6134  6142  6148  6154  6160  6169  6179
```

```
CLC      6212
CLR       926   930   944   945   998  1022  1092  1095  1116  1121  1212  1222  1253  1312  1350
         1351  1410  1453  1457  1493  1506  1546  1550  1576  1599  1631  1715  1789  1886  1912
         1914  1960  1967  2030  2066  2150  2163  2164  2234  2247  2248  2320  2334  2338  2406
         2420  2421  2492  2508  2509  2590  2594  2595  2676  2693  2694  2775  2792  2793  2874
         2890  2892  2973  2990  2991  3072  3089  3090  3171  3188  3189  3270  3287  3288  3369
         3386  3387  3465  3472  3488  3529  3530  3580  3587  3610  3623  3624  3688  3695  3720
         3736  3736  3800  3807  3831  3846  3847  3911  3918  3942  3957  3958  4022  4029  4053
         4068  4069  4133  4140  4164  4179  4180  4249  4304  4305  4368  4429  4494  4561  4585
         4606  4612  4613  4668  4675  4699  4725  4746  4752  4753  4808  4815  4841  4850  4851
         4866  4887  4893  4894  4949  4956  4982  5008  5029  5035  5036  5089  5109  5110  5116
         5130  5131  5137  5210  5230  5231  5237  5251  5252  5258  5314  5315  5391  5405  5457
         5619  5678  5681  5796  5835  5906  5907  5908  5909  5911  5912  5989  6007  6027  6065
         6088  6168  6202  6207  6219
CLRB     5390  5542  5569  5707
CMP       931   955   969  1035  1048  1054  1060  1066  1072  1078  1084  1171  1173  1180  1183
         1219  1314  1363  1377  1390  1411  1726  1732  1902  1904  1997  2076  2079  2092  2094
         2159  2162  2178  2180  2243  2246  2262  2264  2330  2333  2349  2351  2416  2419  2435
         2437  2491  2504  2507  2523  2525  2590  2593  2609  2611  2689  2692  2708  2710  2788
         2791  2807  2809  2887  2890  2906  2908  2986  2989  3005  3007  3085  3088  3104  3106
         3184  3187  3203  3205  3283  3286  3302  3304  3382  3385  3401  3403  3506  3511  3515
         3517  3543  3545  3619  3622  3638  3640  3731  3734  3750  3752  3842  3845  3861  3863
         3953  3956  3972  3974  4064  4067  4083  4085  4175  4178  4194  4196  4287  4302  4318
         4320  4597  4599  4627  4629  4737  4739  4767  4769  4878  4880  4908  4910  5020  5022
         5050  5052  5106  5128  5152  5155  5158  5163  5166  5169  5172  5227  5249  5273  5276
         5279  5284  5287  5290  5293  5372  5398  5452  5699  5863  5879  5882  5890  5964  6009
         6019  6048  6074
CMPB     5380  5384  5535  5537  5545  5566  5570  6120  6126  6136  6143  6149  6156  6162
COM      1907
DEC      1736  5318  5838  6095
DECB     5549  5552  5627  5638
EMT        84
HALT      484  1090  1677  1680  1681  1684  1685  1688  1689  1692  1693  1696  1697  1700  1701
         1978  5444  5454  5525  5789  5813  6026  6171
INC       967   983  5316  5397  5433  5633  5641  5685  5797  5841  5992  6176
INCB     5402  5427  5572
IOT        85
JMP       488  1039  1047  1093  1293  1392  1728  1734  1739  1742  1747  1787  2018  4285  5095
         5154  5160  5165  5175  5216  5275  5281  5286  5296  5340  5883  5893  5913  6030  6050
         6093  6113
JSR      5092  5213  5331  5335  5439  5544  5551  5558  5891  6129  6147  6159  6165  6228
MOV       929   933   935   936   937   938   939   940   941   942   943   947   948   951   952
         953   954   959   961   962   963   986   988   989   990   991   993   996   997  1001
        1002  1013  1016  1019  1020  1021  1024  1025  1027  1028  1038  1046  1050  1052  1053
        1056  1062  1068  1074  1080  1086  1108  1111  1115  1118  1122  1127  1128  1130  1131
        1135  1137  1155  1158  1165  1166  1169  1170  1176  1177  1178  1182  1185  1200  1204
        1211  1213  1215  1216  1217  1225  1226  1227  1229  1243  1246  1254  1255  1257  1258
        1265  1268  1269  1270  1271  1278  1281  1282  1283  1284  1291  1299  1304  1309  1316
        1322  1340  1343  1352  1353  1354  1362  1366  1367  1368  1369  1376  1379  1380  1381
        1382  1389  1397  1402  1407  1413  1418  1437  1440  1443  1448  1450  1452  1454  1456
        1459  1466  1467  1470  1472  1479  1481  1490  1491  1494  1496  1501  1502  1504  1530
        1533  1536  1541  1543  1545  1547  1549  1552  1559  1560  1563  1565  1572  1574  1593
        1584  1587  1589  1594  1595  1597  1621  1624  1627  1632  1633  1634  1639  1640  1641
        1642  1644  1648  1649  1650  1651  1663  1682  1686  1690  1694  1698  1702  1706  1708
        1709  1710  1711  1712  1714  1716  1717  1720  1723  1724  1725  1741  1744  1783  1785
        1786  1812  1815  1832  1835  1838  1842  1947  1849  1852  1863  1864  1870  1879  1880
```

1906	1911	1913	1940	1943	1949	1957	1959	1961	1962	1965	1966	1988	1992	1993
1995	2001	2002	2004	2017	2020	2021	2022	2023	2026	2031	2047	2050	2058	2060
2061	2062	2065	207[illegible]	2073	2080	2085	2086	2087	2089	2099	2101	2102	2103	2104
2105	2107	2108	2123	2126	2134	2135	2136	2137	2146	2153	2156	2165	2170	2171
2172	2175	2185	2187	2188	2189	2190	2191	2194	2195	2208	2211	2219	2220	2221
2222	2231	2237	2240	2249	2254	2255	2256	2259	2269	2271	2272	2273	2274	2275
2278	2279	2292	2295	2303	2304	2305	2306	2316	2324	2327	2336	2341	2342	2343
2346	2356	2358	2359	2360	2361	2362	2365	2366	2379	2382	2390	2391	2392	2393
2403	2410	2413	2422	2427	2428	2429	2432	2442	2444	2445	2446	2447	2448	2451
2462	2465	2468	2471	2472	2473	2474	2475	2490	2498	2501	2510	2515	2516	2517
2520	2530	2532	2533	2534	2535	2536	2539	2540	2553	2556	2564	2565	2566	2567
2576	2584	2587	2596	2601	2602	2603	2606	2616	2618	2619	2620	2621	2622	2625
2626	2643	2646	2654	2655	2656	2657	2660	2669	2669	2675	2683	2686	2695	2700
2701	2702	2705	2715	2717	2718	2719	2720	2721	2724	2725	2742	2745	2753	2754
2755	2756	2759	2767	2768	2774	2782	2785	2794	2799	2800	2801	2804	2814	2816
2817	2818	2819	2820	2823	2824	2841	2844	2852	2853	2854	2855	2858	2866	2867
2873	2881	2884	2893	2898	2899	2900	2903	2913	2915	2916	2917	2918	2919	2922
2923	2940	2943	2951	2952	2953	2954	2957	2965	2966	2972	2980	2983	2992	2997
2998	2999	3002	3012	3014	3015	3016	3017	3018	3021	3022	3039	3042	3050	3051
3052	3053	3056	3064	3065	3071	3079	3082	3091	3096	3097	3098	3101	3111	3113
3114	3115	3116	3117	3120	3121	3138	3141	3149	3150	3151	3152	3155	3163	3164
3170	3178	3181	3190	3195	3196	3197	3200	3210	3212	3213	3214	3215	3216	3219
3220	3237	3240	3248	3249	3250	3251	3254	3262	3263	3269	3277	3280	3289	3294
3295	3296	3299	3309	3311	3312	3313	3314	3315	3318	3319	3336	3339	3347	3348
3349	3350	3353	3361	3362	3368	3376	3379	3388	3393	3394	3395	3398	3408	3410
3411	3412	3413	3414	3417	3418	3438	3441	3450	3451	3459	3461	3463	3464	3466
3468	3469	3471	3473	3477	3478	3480	3481	3483	3484	3486	3501	3503	3516	3518
3519	3520	3521	3522	3524	3531	3536	3537	3538	3540	3562	3565	3574	3576	3578
3579	3581	3583	3584	3586	3588	3592	3593	3595	3596	3597	3598	3599	3608	3613
3616	3625	3630	3631	3632	3635	3645	3647	3648	3649	3650	3651	3654	3655	3671
3674	3684	3686	3687	3689	3691	3692	3694	3696	3700	3701	3703	3704	3705	3706
3707	3708	3718	3725	3728	3737	3742	3743	3744	3747	3757	3759	3760	3761	3762
3763	3766	3767	3783	3786	3796	3798	3799	3801	3803	3804	3806	3808	3812	3813
3815	3816	3817	3818	3819	3820	3830	3836	3839	3848	3853	3854	3855	3856	3868
3870	3871	3872	3873	3874	3877	3878	3894	3897	3907	3909	3910	3912	3914	3915
3917	3919	3923	3924	3926	3927	3928	3929	3930	3931	3941	3947	3950	3959	3964
3965	3966	3969	3979	3981	3982	3983	3984	3985	3988	3989	4005	4008	4018	4020
4021	4023	4025	4026	4028	4030	4034	4035	4037	4038	4039	4040	4041	4042	4052
4058	4061	4070	4075	4076	4077	4080	4090	4092	4093	4094	4095	4096	4099	4100
4116	4119	4129	4131	4132	4134	4136	4137	4139	4141	4145	4146	4143	4149	4150
4151	4152	4153	4163	4169	4172	4181	4186	4187	4188	4191	4201	4203	4204	4205
4206	4207	4210	4211	4229	4232	4235	4243	4244	4245	4246	4247	4250	4251	4252
4253	4254	4255	4256	4258	4259	4260	4262	4264	4265	4266	4278	4279	4291	4293
4294	4295	4296	4297	4299	4306	4311	4312	4313	4315	4336	4339	4347	4349	4350
4351	4360	4373	4375	4376	4377	4378	4396	4399	4407	4408	4416	4417	4423	4425
4426	4428	4438	4440	4441	4442	4443	4461	4464	4472	4473	4481	4482	4488	4490
4491	4493	4503	4505	4506	4507	4508	4534	4537	4545	4548	4549	4550	4559	4564
4566	4572	4573	4574	4583	4588	4590	4603	4604	4605	4607	4608	4614	4619	4620
4621	4624	4651	4654	4664	4666	4667	4669	4671	4672	4674	4676	4680	4681	4683
4686	4687	4688	4697	4702	4704	4710	4711	4712	4713	4714	4723	4728	4730	4743
4744	4745	4747	4748	4754	4759	4760	4761	4764	4791	4794	4804	4806	4807	4809
4811	4812	4814	4816	4820	4821	4823	4826	4827	4828	4829	4830	4839	4844	4846
4853	4854	4855	4864	4869	4871	4884	4885	4886	4888	4889	4895	4900	4901	4902
4905	4932	4935	4945	4947	4948	4950	4952	4953	4955	4957	4961	4962	4964	4967
4968	4969	4970	4971	4980	4985	4987	4993	4994	4995	4996	4997	5006	5011	5013
5026	5027	5028	5030	5031	5037	5042	5043	5044	5047	5080	5083	5086	5088	5091

```
            5097  5098  5099  5103  5104  5105  5114  5115  5119  5120  5121  5122  5124  5125  5127
            5135  5136  5159  5170  5201  5204  5207  5209  5212  5218  5219  5220  5224  5225  5226
            5235  5236  5240  5241  5242  5243  5245  5246  5248  5256  5257  5280  5291  5321  5325
            5328  5330  5332  5367  5368  5370  5373  5378  5388  5400  5401  5403  5404  5407  5408
            5429  5434  5447  5450  5456  5478  5479  5480  5481  5482  5483  5484  5485  5486  5487
            5494  5495  5496  5497  5498  5499  5500  5501  5502  5503  5527  5528  5532  5547  5602
            5610  5611  5612  5618  5625  5643  5644  5645  5646  5647  5668  5669  5670  5671  5672
            5673  5674  5679  5682  5702  5708  5709  5710  5711  5712  5714  5715  5731  5732  5736
            5742  5743  5778  5779  5780  5781  5782  5783  5784  5785  5786  5787  5788  5794  5795
            5799  5800  5801  5802  5803  5804  5805  5806  5807  5810  5829  5830  5831  5832  5833
            5834  5837  5842  5860  5861  5862  5868  5869  5872  5874  5875  5876  5877  5878  5888
            5902  5903  5904  5905  5910  5988  6005  6013  6014  6015  6016  6018  6025  6061  6062
            6063  6064  6066  6075  6077  6078  6087  6091  6098  6104  6110  6111  6114  6117  6122
            6128  6131  6138  6145  6146  6151  6164  6166  6181  6182  6200  6201  6218  6225  6226
            6227  6230  6234
MOVB         946  1018  1110  1160  1206  1248  1345  1442  1535  1626  1837  1945  2052  2128  2213
            2297  2384  2470  2558  2648  2747  2846  2945  3044  3143  3242  3341  3443  3567  3676
            3788  3899  4010  4121  4234  4341  4401  4466  4539  4656  4796  4937  5085  5206  5406
            5436  5529  5557  5565  5603  5604  5607  5608  5609  5613  5616  5617  5636  5677  5680
            5694  5697  5706  5734  5836  5889  6089
NEG         5614  5676
NOP         1030  1117  1168  1214  1261  1262  1263  1264  1274  1275  1276  1277  1287  1288  1289
            1290  1358  1359  1360  1361  1372  1373  1374  1375  1385  1386  1387  1388  1484  1577
            1678  1901  1964  1986  2064  2145  2230  2314  2315  2321  2401  2402  2407  2489  2495
            2575  2577  2581  2674  2773  2872  2971  3070  3169  3268  3367  3485  3607  3716  3717
            3721  3828  3829  3939  3940  4050  4051  4161  4162  4274  4359  4361  4369  4427  4430
            4492  4495  4558  4562  4582  4586  4696  4700  4722  4726  4838  4842  4863  4867  4979
            4983  5005  5009  5126  5247  5336  5337  5338
RESET       5334
ROL         5620  5622  5623  5624  5626  6069
ROR         5846  5847  5848  5849  5850  5851  6213  6214  6215
RTI          960  5409  5458  5488  5504  5534  5648  5716  5744  5812  5932  5939  5946  5953  5960
            6047  6079
RTS         5574  5737  5844  5995  6022  6183  6236
SOB          992  1023  1221  1260  1273  1286  1357  1371  1384  1646  1661  1721  1729  3476  3591
            3699  3811  3922  4033  4144  4248  4679  4819  4960  6017  6070
SUB         5435  5683
TRAP        5746  5755  5756  5757  5758  5761  5762  5764  5765  5766  5767  5768  5769  5770  5771
            5772
TST         1031  1051  1119  1123  1259  1272  1285  1320  1356  1370  1383  1460  1461  1473  1474
            1485  1497  1553  1554  1566  1567  1578  1590  1643  1645  1655  1659  1810  1813  1853
            1857  1871  1873  2024  2081  2100  2147  2166  2186  2232  2250  2270  2317  2337  2357
            2404  2423  2443  2494  2511  2531  2578  2597  2617  2661  2662  2696  2716  2760  2761
            2795  2815  2859  2860  2894  2914  2958  2959  2993  3013  3057  3058  3092  3112  3156
            3157  3191  3211  3255  3256  3290  3310  3354  3355  3389  3409  3532  3609  3626  3646
            3719  3738  3758  3849  3869  3960  3980  4071  4091  4182  4202  4275  4292  4307  4362
            4374  4409  4410  4439  4474  4475  4504  4560  4584  4615  4698  4724  4755  4840  4865
            4896  4981  5007  5038  5369  5395  5442  5448  5531  5539  5561  5631  5688  5698  5733
            5873  5928  5930  5937  5944  5951  5958  6177  6203
TSTB        5382  5523  5563  5690  5704  6115
.ASCII       583   584  6259  6269  6289  6302  6328  6335  6338  6348  6360  6374  6387  6399  6413
            6449  6504  6604  6611  6623  6630  6642  6658  6674  6691  6708  6717  6734  6743  6757
            6766  6775  6783  6789  6812  6821  6838  6846  6854  6869  6877  6885  6900  6908  6926
            6934  6943  6959  6967  6976  6992  7012  7030  7047  7054  7073  7083  7097  7105  7113
            7122  7137  7145  7161  7167  7191  7200  7219  7227  7234  7239  7247  7261  7269  7276
            7281  7289  7303  7311  7318  7323  7332  7346  7354  7361  7366  7374  7420  7438  7443
```

```
                7458    7477    7484    7519    7525    7555    7577    7590    7599    7622    7660    7685    7702    7712    7726
.ASCIZ           582     585     974    5343    6240    6242    6249    6257    6262    6277    6279    6283    6288    6292    6308
                6311    6316    6320    6323    6345    6355    6370    6383    6386    6409    6422    6426    6429    6436    6442
                6454    6462    6468    6474    6479    6485    6491    6497    6512    6516    6521    6527    6537    6543    6548
                6554    6560    6565    6571    6579    6590    6597    6620    6639    6649    6665    6681    6698    6724    6751
                6798    6802    6829    6861    6892    6917    6951    6984    7000    7019    7036    7062    7092    7130    7151
                7176    7182    7206    7254    7296    7339    7381    7387    7429    7452    7466    7492    7499    7511    7528
                7531    7536    7560    7568    7584    7596    7607    7627    7636    7644    7653    7665    7670    7679    7691
                7695    7708    7718    7732    7739
.BLKB           5865
.BLKW           5721
.BYTE            510     511     516     517     525     526     534     535     536     537    5342    5649    5650    5651    5652
                6140    6141    7400    7749    7752    7755    7757    7770    7791    7796    7804    7814    7817    7822    7826
                7829    7833    7837    7846    7850    7855
.ENABL             1      45
.END            7984
.ENDC            544      71      73      74      75     462     489     493     497     499     504     508     510     538     560
                 580     581     582     583     587     609     612     615     618     621     624     627     630     633     636
                 639     642     645     648     651     654     657     660     663     666     669     672     675     678     682
                 685     688     691     694     697     700     703     706     709     712     715     718     721     724     727
                 730     733     736     739     743     746     749     752     755     758     761     764     767     770     773
                 776     779     782     785     788     792     795     798     801     804     807     810     813     816     819
                 822     825     828     831     835     838     841     844     847     850     853     856     859     862     865
                 868     871     874     878     881     884     887     890     893     896     899     902     905     908     911
                 914     917     920     923     933     934     937     939     941     943     944     945     947     949     965
                 969     971     972     974    1006    1007    1011    1012    1013    1014    1018    1098    1099    1104    1105
                1106    1110    1113    1114    1115    1142    1143    1153    1154    1155    1156    1160    1164    1165    1191
                1192    1198    1199    1200    1201    1206    1209    1210    1235    1236    1241    1242    1243    1244    1248
                1251    1252    1253    1327    1328    1338    1339    1340    1341    1345    1348    1349    1350    1423    1424
                1435    1436    1437    1438    1442    1445    1446    1447    1516    1517    1528    1529    1530    1531    1535
                1538    1539    1540    1609    1610    1619    1620    1621    1622    1626    1629    1630    1631    1792    1793
                1807    1808    1809    1819    1820    1830    1831    1832    1833    1837    1840    1841    1842    1858    1864
                1867    1874    1880    1883    1925    1926    1938    1939    1940    1941    1945    1947    1948    1949    2037
                2038    2045    2046    2047    2048    2052    2054    2055    2056    2057    2058    2115    2116    2121    2122
                2123    2124    2128    2130    2131    2132    2133    2134    2200    2201    2206    2207    2208    2209    2213
                2215    2216    2217    2218    2219    2284    2285    2290    2291    2292    2293    2297    2299    2300    2301
                2302    2303    2371    2372    2377    2378    2379    2380    2384    2386    2387    2388    2389    2390    2457
                2458    2463    2464    2465    2466    2470    2545    2546    2551    2552    2553    2554    2558    2560    2561
                2562    2563    2564    2632    2633    2641    2642    2643    2644    2648    2650    2651    2652    2653    2654
                2669    2673    2731    2732    2740    2741    2742    2743    2747    2749    2750    2751    2752    2753    2768
                2772    2830    2831    2839    2840    2841    2842    2846    2848    2849    2850    2851    2852    2867    2871
                2929    2930    2938    2939    2940    2941    2945    2947    2948    2949    2950    2951    2966    2970    3028
                3029    3037    3038    3039    3040    3044    3046    3047    3048    3049    3050    3065    3069    3127    3128
                3136    3137    3138    3139    3143    3145    3146    3147    3148    3149    3164    3168    3226    3227    3235
                3236    3237    3238    3242    3244    3245    3246    3247    3248    3263    3267    3325    3326    3334    3335
                3336    3337    3341    3343    3344    3345    3346    3347    3362    3366    3427    3428    3436    3437    3438
                3439    3443    3445    3446    3447    3448    3449    3551    3552    3560    3561    3562    3563    3567    3569
                3570    3571    3572    3573    3660    3661    3669    3670    3671    3672    3676    3678    3679    3680    3681
                3682    3772    3773    3781    3782    3783    3784    3788    3790    3791    3792    3793    3794    3883    3884
                3892    3893    3894    3895    3899    3901    3902    3903    3904    3905    3994    3995    4003    4004    4005
                4006    4010    4012    4013    4014    4015    4016    4105    4106    4114    4115    4116    4117    4121    4123
                4124    4125    4126    4127    4216    4217    4227    4228    4229    4230    4234    4237    4238    4239    4240
                4241    4326    4327    4334    4335    4336    4337    4341    4343    4344    4345    4346    4347    4386    4387
                4394    4395    4396    4397    4401    4403    4404    4405    4406    4407    4417    4421    4451    4452    4459
                4460    4461    4462    4466    4468    4469    4470    4471    4472    4482    4486    4519    4520    4532    4533
                4534    4535    4539    4541    4542    4543    4544    4545    4636    4637    4649    4650    4651    4652    4656
```

F16


```
                4658    4659    4660    4661    4662    4776    4777    4789    4790    4791    4792    4796    4798    4799    4800
                4801    4802    4917    4918    4930    4931    4932    4933    4937    4939    4940    4941    4942    4943    5059
                5060    5078    5079    5080    5081    5085    5088    5180    5181    5199    5200    5201    5202    5206    5209
                5306    5308    5309    5311    5314    5320    5323    5324    5328    5330    5332    5334    5340    5342    5343
                5346    5350    5353    5358    5362    5364    5375    5378    5381    5382    5384    5386    5393    5397    5402
                5403    5407    5410    5411    5415    5418    5427    5434    5439    5440    5441    5442    5452    5456    5459
                5463    5509    5529    5580    5658    5726    5732    5735    5754    5755    5756    5757    5758    5759    5760
                5761    5762    5763    5764    5765    5766    5767    5768    5769    5770    5771    5772    5777    5786    5787
                5793    5799    5800    5810    5812    5816    5820
.EQUIV           84      85      87     109     110     111     112     113     114     116     117     118     148     149     150
                151     152     153     154     155     156     157     176     177     179     179     180     181     182     183
                184     186     235     236     237     238     445     446     447     448     449     450     451     452     453
                454     455     456     457     458     459     460
.EVEN           974    7409    7858
.IF              50      71      72      73      74      75     462     487     492     495     497     503     507     509     538
                560     580     581     582     586     587     607     610     613     616     619     622     625     628     631
                634     637     640     643     646     649     652     655     658     661     664     667     670     673     676
                680     683     686     689     692     695     698     701     704     707     710     713     716     719     722
                726     728     731     734     737     740     743     746     749     752     755     758     761     764     767
                770     773     776     779     782     785     790     793     796     799     802     805     808     811     814
                817     820     823     826     829     833     836     839     842     845     848     851     854     857     860
                863     866     869     872     876     879     882     885     888     891     894     897     900     903     906
                909     912     915     918     921     928     933     935     937     939     941     943     944     945     947
                965     968     969     970     972     973    1005    1007    1011    1013    1014    1016    1097    1099    1104
               1106    1108    1113    1114    1115    1141    1143    1153    1155    1156    1158    1163    1164    1165    1190
               1192    1198    1200    1201    1204    1209    1210    1234    1236    1241    1243    1244    1246    1251    1252
               1326    1328    1338    1340    1341    1343    1348    1349    1422    1424    1435    1437    1438    1440    1445
               1446    1515    1517    1528    1530    1531    1533    1538    1539    1608    1610    1619    1621    1622    1624
               1629    1630    1791    1793    1807    1809    1818    1820    1830    1832    1833    1835    1840    1841    1857
               1864    1867    1873    1880    1883    1924    1926    1938    1940    1941    1943    1947    1948    1949    2036
               2038    2045    2047    2048    2050    2054    2055    2056    2057    2114    2116    2121    2123    2124    2126
               2130    2131    2132    2133    2199    2201    2206    2208    2209    2211    2215    2216    2217    2218    2283
               2285    2290    2292    2293    2295    2299    2300    2301    2302    2370    2372    2377    2379    2380    2382
               2386    2387    2388    2389    2456    2458    2463    2465    2466    2468    2544    2546    2551    2553    2554
               2556    2560    2561    2562    2563    2631    2633    2641    2643    2644    2646    2650    2651    2652    2653
               2669    2672    2730    2732    2740    2742    2743    2745    2749    2750    2751    2752    2768    2771    2829
               2831    2839    2841    2842    2844    2848    2849    2850    2851    2867    2870    2928    2930    2938    2940
               2941    2943    2947    2948    2949    2950    2966    2969    3027    3029    3037    3039    3040    3042    3046
               3047    3048    3049    3065    3068    3126    3128    3136    3138    3139    3141    3145    3146    3147    3148
               3164    3167    3225    3227    3235    3237    3238    3240    3244    3245    3246    3247    3263    3266    3324
               3326    3334    3336    3337    3339    3343    3344    3345    3346    3362    3365    3426    3428    3436    3438
               3439    3441    3445    3446    3447    3448    3550    3552    3560    3562    3563    3565    3569    3570    3571
               3572    3659    3661    3669    3671    3672    3674    3678    3679    3680    3681    3771    3773    3781    3783
               3784    3786    3790    3791    3792    3793    3882    3884    3892    3894    3895    3897    3901    3902    3903
               3904    3993    3995    4003    4005    4006    4008    4012    4013    4014    4015    4104    4106    4114    4116
               4117    4119    4123    4124    4125    4126    4215    4217    4227    4229    4230    4232    4237    4238    4239
               4240    4325    4327    4334    4336    4337    4339    4343    4344    4345    4346    4385    4387    4394    4396
               4397    4399    4403    4404    4405    4406    4417    4420    4450    4452    4459    4461    4462    4464    4468
               4469    4470    4471    4482    4485    4518    4520    4532    4534    4535    4537    4541    4542    4543    4544
               4635    4637    4649    4651    4652    4654    4658    4659    4660    4661    4775    4777    4789    4791    4792
               4794    4798    4799    4800    4801    4916    4918    4930    4932    4933    4935    4939    4940    4941    4942
               5058    5060    5078    5080    5081    5083    5088    5179    5181    5199    5201    5202    5204    5209    5305
               5306    5307    5308    5309    5310    5311    5313    5319    5322    5324    5328    5330    5332    5340    5342
               5343    5349    5352    5357    5362    5374    5376    5377    5378    5382    5383    5384    5393    5395    5403
               5404    5409    5410    5411    5414    5417    5427    5430    5437    5439    5440    5442    5445    5452    5456
               5459    5462    5508    5529    5579    5657    5725    5731    5735    5746    5755    5756    5757    5758    5759
```

```
            5760    5761    5762    5763    5764    5765    5766    5767    5768    5769    5770    5771    5772    5775    5786
            5787    5792    5799    5800    5809    5910    5812    5816    5819
.IFF           71      73      74      75     493     497     499     504     507     509     538     587     607     612     615
             618     621     624     627     630     633     636     639     640     643     648     651     654     657     660
             663     666     669     672     675     678     682     685     688     691     694     697     700     703     706
             709     712     715     718     721     724     727     730     733     736     739     740     743     746     749
             752     755     758     761     764     767     770     773     776     779     782     785     790     793     796
             801     804     807     810     813     816     819     822     825     828     831     835     838     841     844
             845     848     851     854     857     860     865     868     863     872     876     879     882     885     888
             893     896     897     900     903     906     911     914     917     920     921     933     968     970    1006
            1007    1012    1013    1018    1098    1099    1105    1106    1110    1142    1143    1154    1155    1160    1191
            1192    1199    1200    1206    1235    1236    1242    1243    1248    1327    1328    1339    1340    1345    1423
            1424    1436    1437    1443    1516    1517    1529    1530    1535    1609    1610    1620    1621    1626    1792
            1793    1808    1809    1819    1820    1831    1832    1937    1857    1873    1925    1926    1939    1940    1945
            2037    2038    2046    2047    2052    2115    2116    2122    2123    2128    2200    2201    2207    2208    2213
            2284    2285    2291    2292    2297    2371    2372    2378    2379    2384    2457    2458    2464    2465    2470
            2545    2546    2552    2553    2558    2632    2633    2642    2643    2648    2731    2732    2741    2742    2747
            2830    2831    2840    2841    2846    2929    2930    2939    2940    2945    3028    3029    3038    3039    3044
            3127    3128    3137    3138    3143    3226    3227    3236    3237    3242    3325    3326    3335    3336    3341
            3427    3428    3437    3438    3443    3551    3552    3561    3562    3567    3660    3661    3670    3671    3676
            3772    3773    3782    3783    3788    3883    3884    3893    3894    3899    3994    3995    4004    4005    4010
            4105    4106    4115    4116    4121    4216    4217    4228    4229    4234    4326    4327    4335    4336    4341
            4386    4387    4395    4396    4401    4451    4452    4460    4461    4466    4519    4520    4533    4534    4539
            4636    4637    4650    4651    4656    4776    4777    4790    4791    4796    4917    4918    4931    4932    4937
            5059    5060    5079    5080    5085    5180    5181    5200    5201    5206    5306    5310    5314    5319    5322
            5342    5350    5375    5378    5384    5410    5415    5417    5430    5452    5456    5463    5509    5580    5658
            5726    5732    5777    5793    5808    5820
.IFT         974    5392    5440
.IFTF        974    5390    5439
.IIF          49      54      59      60      68      69      70      71      74      75      76     484     586     934     937
             943     944     945     947     948     969    5308    5314    5315    5326    5342    5346    5353    5354    5355
            5356    5357    5358    5362    5391    5392    5407    5410    5411    5418    5419    5420    5421    5422    5427
            5445    5452    5459    5576    5754    5755    5756    5757    5758    5761    5762    5764    5765    5766    5767
            5768    5769    5770    5771    5772
.IRP         462     606     679     740     789     832     875    1005    1097    1141    1190    1234    1326    1422    1515
            1608    1791    1818    1924    2036    2114    2199    2283    2370    2456    2544    2631    2730    2829    2928
            3027    3126    3225    3324    3426    3550    3659    3771    3882    3993    4104    4215    4325    4385    4450
            4518    4635    4775    4916    5058    5179    5330    5456    5478    5498    5668    5708    5780    5786    5799
            5800
.LIST          1       2      74     461     462     484     538     540     541     542     543     544     545     546     547
             548     549     550     551     552     553     554     555     556     557     558     559     560     561     562
             563     564     565     566     567     568     569     570     571     572     573     574     575     576     577
             578     579     580     949     969     972     974    1005    1013    1097    1106    1141    1155    1190    1200
            1234    1243    1326    1340    1422    1437    1515    1530    1608    1621    1791    1809    1818    1832    1924
            1940    2036    2047    2114    2123    2199    2208    2283    2292    2370    2379    2456    2465    2544    2553
            2631    2643    2730    2742    2829    2841    2928    2940    3027    3039    3126    3138    3225    3237    3324
            3336    3426    3438    3550    3562    3659    3671    3771    3783    3882    3894    3993    4005    4104    4116
            4215    4229    4325    4336    4385    4396    4450    4461    4518    4534    4635    4651    4775    4791    4916
            4932    5058    5080    5179    5201    5314    5330    5332    5357    5452    5746    5754    5755    5756    5757
            5758    5759    5761    5762    5763    5764    5765    5766    5767    5768    5769    5770    5771    5772    5773
.MACRO         1       6       7       8      10      12      13      14      15      16      17      18      19      21      22
              23      24      25      26      27      28      29      30      31      32      33      34      35      36      37
              38      39      40      41      42      43      75     501    1004    1097    1141    1189    1233    1326    1421
            1422    1515    1608    1791    1818    1924    2033    2035    2114    2199    2283    2370    2456    2544    2630
            2631    2730    2829    2928    3027    3126    3225    3324    3423    3424    3425    3426    3550    3659    3771
            3882    3993    4104    4215    4325    4383    4385    4450    4515    4516    4517    4518    4635    4775    4916
```

```
            5058    5179    5746    7218
.MCALL        46      47      48     461     949     972
.NLIST         1       3       4     461     462     494     538     540     541     542     543     544     545     546     547
             548     549     550     551     552     553     554     555     556     557     558     559     560     561     562
             563     564     565     566     567     568     569     570     571     572     573     574     575     576     577
             963     964     965     966     969     972     974    1005    1013    1097    1106    1141    1155    1190    1200
            1234    1243    1326    1340    1422    1437    1515    1530    1608    1621    1791    1809    1818    1832    1924
            1940    2036    2047    2114    2123    2199    2208    2283    2292    2370    2379    2456    2465    2544    2553
            2631    2643    2730    2742    2829    2841    2928    2940    3027    3039    3126    3138    3225    3237    3324
            3336    3426    3438    3550    3562    3659    3671    3771    3783    3882    3894    3993    4005    4104    4116
            4216    4228    4325    4336    4385    4396    4450    4461    4518    4534    4635    4651    4775    4791    4916
            4932    5058    5080    5179    5201    5314    5330    5332    5357    5452    5746    5754    5755    5756    5757
            5758    5759    5761    5762    5763    5764    5765    5766    5767    5768    5769    5770    5771    5772    5773
.PAGE        501     587
.REPT        494     540     560
.SBTTL        64      77     203     213     226     374     478     487     490     501     587     927     965    1005    1097
            1141    1190    1234    1326    1422    1515    1608    1791    1818    1924    2036    2114    2199    2283    2370
            2456    2544    2631    2730    2829    2928    3027    3126    3225    3324    3426    3550    3659    3771    3882
            3993    4104    4215    4325    4385    4450    4518    4635    4775    4916    5058    5179    5303    5347    5412
            5460    5606    5577    5655    5723    5746    5774    5817
.TITLE        49
.WORD        484     485     486     498     509     512     513     514     515     518     519     520     521     522     523
             524     527     528     529     538     540     541     542     543     544     545     546     547     548     549
             550     551     552     553     554     555     556     557     558     559     560     561     562     563     564
             566     566     567     568     569     570     571     572     573     574     575     576     577     578     579
             607     610     613     616     619     622     625     628     631     634     637     640     643     646     649
             652     655     658     661     664     667     670     673     676     680     683     686     689     692     695
             698     701     704     707     710     713     716     719     722     725     728     731     734     737     741
             744     747     750     753     756     759     762     765     768     771     774     777     780     783     786
             790     793     796     799     802     805     808     811     814     817     820     823     826     829     833
             836     239     842     845     848     851     854     857     860     863     866     869     872     875     879
             882     885     888     891     894     897     900     903     906     909     912     915     918     921    1041
            1296    1394    1509    1510    1602    1603    1750    1752    1754    1757    1758    1759    1760    1761    1762
            1763    1764    1765    1766    1768    1769    1770    1771    1772    1773    1775    1776    1777    1778    1779
            1780    1919    1920    5148    5149    5269    5270    5319    5322    5341    5573    5653    5753    5809    5811
            5895    5914    5934    5941    5948    5955    5962    5967    5968    5969    5970    5971    5972    5973    5974
            5975    5976    5977    5978    6012    6024    6031    6033    6086    6101    6103    6107    6109    6133    6153
            6168    6174    6185    6191    6192    6232    7410    7862    7865    7868    7871    7884    7887    7890    7893
            7908    7911    7914    7917    7922    7925    7927    7929    7931    7933    7936    7939    7944    7948    7951
            7957    7960    7962    7969    7973    7975    7978    7981

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DEKBCB/NL:TOC/SOL/CRF=SYSMAC.SML,DEKBCB.P11
RUN-TIME: 72 89 16 SECONDS
RUN-TIME RATIO: 341/179=1.9
CORE USED:  42K  (83 PAGES)
```

Spooler runtime 30 Seconds, 126 KCS, 857 disk reads, 0 disk writes, 202 pages