RM03

DISKLESS DIAGNOSTIC
MD-11-DZRMJ-A

EP-DZRMJ-A-DL-A

FICHE 1 OF 2

OCT 1977

digital

MADE IN USA

RM03

DISKLESS DIAGNOSTIC

MD-11-DZRMJ-A

EP-DZRMJ-A-DL-A

OCT 1977

digital

FICHE 2 OF 2

MADE IN USA

```
.REM    \
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRMJ-A-D
PRODUCT NAME:	RM03 DISKLESS DIAGNOSTIC
DATE CREATED:	1 AUGUST 77
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	DOUG RIIKONEN

CONTENTS

```
        6.  XXDP COMPATIBILITY

        7.  OPERATING SYSTEM COMPATIBILITY

    6.  TEST DESCRIPTION
```

1.0 INTRODUCTION

1.1 ABSTRACT

THE RM03 DISKLESS DIAGNOSTIC IS A STAND ALONE PROGRAM WHICH USES FUNCTIONAL AND DIAGNOSTIC MEANS TO VERIFY THE OPERABILITY OF THE RM03 DISK SUBSYSTEM EXCLUDING AND INDEPENDENTLY OF THE STORAGE MODULE DRIVE. IN PARTICULAR, THE PROGRAM SERVES THE FOLLOWING PURPOSES:

TO DETECT ERRORS AND FAULTS IN THE RH70 MASSBUS CONTROLLER;

TO DETECT ERRORS AND FAULTS IN THE RM03 MASSBUS ADAPTER;

TO RESOLVE HARDWARE FAILURES IN THE RH70 AND RM03 TO A FIELD REPLACEABLE MODULE OR MODULES.

1.2 UNIT UNDER TEST

THE UNIT UNDER TEST (UUT) IS THE RM03 DISK SUBSYSTEM, EXCLUDING THE STORAGE MODULE DISK DRIVE AND THE RH70 MASSBUS CONTROLLER.

2.0 OPERATING REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

THE FOLLOWING MINIMUM HARDWARE CONFIGURATION, ASSUMED TO BE OPERATIONAL, IS REQUIRED TO LOAD AND EXECUTE THE RM03 DISKLESS DIAGNOSTIC:

PDP-11 PROCESSOR
24K MEMORY
KW11-L OR KW11-P CLOCK
PROGRAM LOADING DEVICE
TERMINAL
RH CONTROLLER

UNIT UNDER TEST,
WHERE THE UNIT UNDER TEST CONSISTS OF ONE TO EIGHT RM03 ADAPTERS.

NOTE

A STORAGE MODULE DISK DRIVE IS NOT REQUIRED FOR EXECUTING THE RM03 DISKLESS DIAGNOSTIC AS LONG AS THE "DEVICE ADDRESS" AND "PLUG ENABLE" LINES ARE PROVIDED BY SOME OTHER MEANS SUCH AS JUMERS.

```
                                        THE DEVICE ADDRESS LINES  MAY  FLOAT  TO
                                        ADDRESS  7  IF  THERE IS NO OTHER DEVICE
```

WITH ADDRESS 7 ON THE SUBSYSTEM. THE "PLUG ENABLE" LINE MUST BE PROVIDED TO ALLOW MASSBUS DIALOGUE BY THE RM03.

2.2 MEDIA REQUIREMENTS

NONE

2.3 PREREQUISITE DIAGNOSTIC PROGRAMS

NONE

3.0 OPERATING PROCEDURE

3.1 LOADING

THE PROGRAM MAY BE LOADED BY EITHER PAPER TAPE, USING THE STANDARD PAPER TAPE LOADING PROCEDURE, OR XXDP MEDIA, USING THE APPROPRIATE LOADING DEVICE.

3.2 SWITCH OPTIONS

THE FOLLOWING SWITCH OPTIONS ARE INVOKED WHEN THE APPROPRIATE SWITCH IS ON.

SW15 HALT ON ERROR
SW14 LOOP ON TEST (CURRENTLY BEING EXECUTED)
SW13 INHIBIT ERROR TYPEOUTS
SW12 UNUSED
SW11 INHIBIT TEST ITERATIONS
SW10 BELL ON ERROR
SW09 LOOP ON ERROR
SW08 LOOP ON TEST IN SW07-00

THE LOW ORDER 8 SWITCHES ARE USED IN CONJUNCTION WITH SW08 TO SPECIFY A PARTICULAR TEST WHICH THE PROGRAM WILL LOOP ON.

3.3 STARTING

THE PROGRAM STARTS AT LOCATION 200 WHICH USES DEFAULT VALUES OF PARAMETERS AND PROVIDES MAXIMUM TESTING WITH THE SWITCH REGISTER EQUAL TO ZERO.

4.0 OPERATOR INTERFACE

4.1 PROGRAM ID

THE PROGRAM TYPES ITS NAME AND MAINDEC NUMBER THE FIRST TIME IT IS STARTED AFTER BEING LOADED.

4.2 PROGRESS REPORTS

AN END OF PASS REPORT OCCURRS EACH TIME THE PROGRAM IS EXECUTED FOR ALL ADAPTERS IN THE TEST QUE. THE END OF PASS REPORT INCLUDES A MESSAGE AND AN ERROR SUMMARY.

4.3 PERFORMANCE REPORT

NO PERFORMANCE REPORTS ARE GIVEN DURING THE EXECUTION OF THE PROGRAM.

4.4 PROGRAM HALTS

THERE ARE NO SCHEDULED HALTS DURING THE EXECUTION OF THE PROGRAM. PROCESSOR HALTS ARE DUE TO THE TRAP CATCHER.

4.5 ERROR REPORTS

THE RM03 DISKLESS DIAGNOSTIC PROVIDES COMPREHENSIVE ERROR REPORTS INTENDED TO (1) AID IN FAULT RESOLUTION AND (2) MINIMIZE REFERENCES TO PROGRAM LISTINGS.

THE FIRST LINE OF THE ERROR REPORT CONTAINS THE NUMBER OF THE UNIT BEING TESTED, THE TEST NUMBER, THE ERROR NUMBER AND THE VALUE OF THE PROGRAM COUNTER WHERE THE ERROR WAS CALLED. THIS LINE IS FOLLOWED BY THE ERROR MESSAGE: SEVERAL LINES OF TEXT WHICH GIVE A COMPREHENSIVE DESCRIPTION OF THE ERROR, AND A LIST OF FAILING MODULES IN ORDER OF DECREASING PROBABILITY. THE ERROR MESSAGE IS NORMALLY FOLLOWED BY ONE OR MORE PAIRS OF LINES CONTAINING DATA HEADERS AND DATA PERTININENT TO THE ERROR, INCLUDING EXPECTED AND ACTUAL TEST

286 RESULTS.

5.0 ENVIRONMENTAL SUPPORT

5.1 PROCESSOR COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC IS EXECUTABLE ON ANY PDP-11 PROCESSOR, PROVIDING PREVIOUSLY MENTIONED HARDWARE REQUIREMENTS ARE MET.

5.2 DUAL PORT CONFIGURATIONS

THE RM03 DISKLESS DIAGNOSTIC IS NOT EXECUTABLE ON RM03 SUBSYSTEMS HAVING THE DUAL PORT OPTION UNLESS THE DUAL PORT SWITCH IS SET TO THE APPROPRIATE PORT (A OR B) AND NOT TO THE PROGRAMMABLE POSITION (A/B).

5.3 MEMORY PARITY HARDWARE

MEMORY PARITY HARDWARE WILL NOT BE USED DURING THE EXECUTION OF THE RM03 DISKLESS DIAGNGOSTIC.

5.4 MEMORY MANAGEMENT HARDWARE

MEMORY MANAGEMENT HARDWARE WILL NOT BE USED DURING THE RM03 DISKLESS DIAGNOSTIC.

5.5 ACT11, APT11 COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC PROGRAM IS COMPATIBLE WITH ACT11 AND APT11 IN BOTH DUMP AND AUTOMATIC MODES. FURTHER, THE PROGRAM WILL EXECUTE A QUICK PASS DURING THE FIRST PASS IN SUPPORT OF QUICK VERIFY MODE.

5.6 XXDP COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC PROGRAM IS COMPATIBLE WITH XXDP IN DUMP AND CHAIN MODES.

5.7 OPERATING SYSTEM COMPATIBILITY

THE PROGRAM IS NOT REQUIRED TO BE COMPATIBLE WITH ANY OPERATING SYSTEM.

6.0 TEST DESCRIPTION

THE PROGRAM IS DESIGNED IN A BOTTOM UP MANNER SUCH THAT EACH TEST GENERALLY USES A MORE COMPLEX SUBSET OF HARDWARE THAN THE PREVIOUS TEST.

MODULE CALLOUT IS PREDICATED ON THE ASSUMPTION THAT EARLIER TESTS HAVE BEEN COMPLETED WITHOUT ERROR AND THAT ERRORS ARE DUE TO SINGLE, NONTRANSIENT HARDWARE FAILURES.

THE "RM03 DISKLES DIAGNOSTIC" CAN BE EXECUTED USING AN RH70 OR AN RH11 MASSBUS CONTROLLER.

UNLESS SPECIFIED BY THE OPERATOR OR BY THE ENVIRONMENT TABLE THE TEST IS REPEATED FOR EACH POSSIBLE DEVICE STARTING WITH DEVICE 0.

THE MODULES WHICH MAY BE CALLED OUT DURING THE EXECUTION OF THE TEST ARE AS FOLLOWS:

IF
CS
DS
MASSBUS MODULE

THE RADIAL MODULE (RD) IS NOT TESTED BY THIS PROGRAM.

TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE RM03 CAN COMPLETE A REGISTER TRANSFER ON THE MASSBUS, AND, IN PARTICULAR, TO VERIFY THAT "TRANSFER" IS NOT STUCK IN AN INACTIVE STATE.

PROCEDURE:

THE PROGRAM WRITES AND READS REMOTE REGISTERS FOR THE SELECTED DEVICE. REGISTER CONTENTS AND PARITY ERRORS ARE IGNORED, AND THE TEST FAILS IF A "NONEXISTENT DEVICE ERROR" OR BUS TIMEOUT OCCURRS FOR EVERY REGISTER ACCESS. IF THE TEST FAILS THE PROGRAM JUMPS TO THE END OF PASS HANDLER WHICH SELECTS THE NEXT DEVICE TO BE TESTED.

PROBABLE FAULT:

THE TEST FAILS IF THE SELECTED DEVICE IS NONEXISTENT OR IS SWITCHED TO THE PROGRAMMABLE POSITION OR TO THE ALTERNATE PORT. THE FOLLOWING FAULTS ARE APPLICABLE ONLY WHEN THE DEVICE IS PRESENT AND IS SWITCHED TO THE APPROPRIATE PORT.

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

3. CS MODULE

CTOD TEST

PURPOSE:

TO VERIFY THAT DATA CAN BE TRANSFERRED TO AND FROM THE RM03 USING THE CONTROL BUS AND, IN PARTICULAR, TO VERIFY THAT "CONTROLLER TO DEVICE" HAS NOT FAILED.

PROCEDURE:

THE TEST WRITES ONES IN REMOTE REGISTERS THEN READS EACH REGISTER WHICH WILL WRITE ZEROS IN THE REGISTER IF "IF3 CTOD HOLD H" IS STUCK AT ONE. THE TEST THEN READS AS MANY REMOTE REGISTERS AS ARE NECESSARY TO OBTAIN ONE OR MORE ONE BITS.

PROBABLE FAULT:

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

MASSBUS INITIALIZE TEST

PURPOSE:

TO VERIFY THAT THE MASSBUS ADAPTER IS BEING INITIALIZED BY THE MASS BUS.

PROCEDURE:

USING CONTROLLER CLEAR TO INITIALIZE THE SELECTED UNIT, THIS TEST THEN READS MASSBUS ADAPTER REGISTERS TO VERIFY THAT AT LEAST ONE BIT IS CLEARED. MASSBUS ADAPTER REGISTERS ARE PRESET TO A NON ZERO VALUE PRIOR TO CONTROLLER CLEAR.

PROBABLE FAULT:

1. ASYNCHRONOUS MASSBUS MODULE

2. IF MODULE

3. CS MODULE

CLEAR STUCK ACTIVE TEST

PURPOSE:

TO VERIFY THAT "MBA CLR L" ON THE CS MODULE IS NOT STUCK IN AN ACTIVE STATE.

PROCEDURE:

CONTROLLER CLEAR IS USED TO INITIALIZE THE SELECTED UNIT, AFTER WHICH 1'S ARE WRITTEN IN ERROR REGISTERS 1 AND 2 AND MAINTENANCE REGISTER 1. IF ANY 1 BITS CAN BE READ BACK THE TEST IS OK, ELSE, "MBA CLR L" IS PROBABLY STUCK ACTIVE.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

3. ASYNCHRONOUS MASSBUS MODULE

TRISTATE TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE PATH TO AND FROM THE MASSBUS ADAPTER TRI-STATE REGISTER BUS IS NOT STUCK AT ONE OR ZERO AND THAT EACH BIT POSITION IS INDEPENDENT.

PROCEDURE:

THIS TEST PRESETS MASSBUS ADAPTER REGISTERS TO A NONZERO VALUE, THEN, ASSUMING THE REGISTERS ARE PRESET, IT CLEARS THEM USING A MOVE INSTRUCTION. THE TEST THEN READS AS MANY REGISTERS AS IS NECESSARY TO OBTAIN ONE OR MORE ZEROS FROM EACH BIT POSITION.

509

THE TEST CLEARS MASSBUS ADAPTER REGISTERS, THEN, ASSUMING THE REGISTERS ARE CLEARED, IT LOADS THEM WITH ONES AND READS AS MANY REGISTERS AS IS NECESSARY TO OBTAIN ONE OR MORE ONE BITS IN EACH BIT POSITION.

FINALLY, THE TEST WRITES A SINGLE ONE BIT PATTERN IN BIT 0 OF SELECTED REMOTE REGISTERS AND VERIFIES THAT THE PATTERN CAN BE READ BACK. THE ONE BIT IS SHIFTED AND THE TEST REPEATED FOR ALL BIT POSITIONS.

PROBABLE FAULT:

1. ASYNCHRONOUS MASSBUS MODULE

2. IF MODULE

3. CS MODULE

4. DS MODULE

REGISTER SELECT TEST

PURPOSE:

TO VERIFY THAT THE REGISTER SELECT LINES ARE NOT IN A STUCK POSITION.

PROCEDURE:

EACH REGISTER SELECT LINE IS TESTED BY WRITING ZEROS IN THOSE DEVICE REGISTERS FOR WHICH THE LINE MUST BE ZERO, THEN WRITING ONES IN THOSE DEVICE REGISTERS FOR WHICH THE LINE MUST BE ONE. THE ZERO REGISTER IS READ BACK AND IF THE SELECT LINE IS STUCK AT ZERO, THE ZERO REGISTER WILL CONTAIN ONES. THE PROCESS IS REPEATED TO DETECT A STUCK AT ONE FAULT, EXCEPT IN THIS CASE, THE ONES REGISTER IS WRITTEN FIRST.

REGISTER SELECT LINES 1, 2, 4 AND 8 ARE TESTED IN THIS MANNER; SELECT LINE 16 IS EXPLICITLY TESTED IN THE "ILR TEST".

PROBABLE FAULT:

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

DRIVE TYPE TEST

PURPOSE:

TO TEST THE "DRIVE TYPE" REGISTER, RMDT.

PROCEDURE:

THE PROGRAM READS RMDT AND VERIFIES THAT THE RESULT CORRESPONDS TO A SINGLE PORT OR DUAL PORT RM03 DRIVE.

PROBABLE FAULT:

1. IF MODULE

DEVICE AVAILABLE TEST

PURPOSE:

TO VERIFY THAT DEVICE AVAILABLE STATUS IS SET.

PROCEDURE:

THE PROGRAM TESTS "DVA", BIT 11 OF RMCS1.

PROBABLE FAULT:

1. IF MODULE

HOLDING REGISTER TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE HOLDING REGISTER IS NOT STUCK AT ONE, STUCK AT ZERO, AND THAT THERE IS NO BIT INTERFERENCE.

PROCEDURE:

THE PROGRAM TRANSFERS ONES, THEN ZEROS TO THE HOLDING REGISTER AND VERIFIES THAT NONE OF THE BITS ARE STUCK AT ONE.

THE PROGRAM TRANSFERS ZEROS, THEN ONES TO THE HOLDING REGISTER AND VERIFIES THAT NONE OF THE BITS ARE STUCK AT ZERO.

FINALLY, THE TEST TRANSFERS A SHIFTING ONE BIT PATTERN AND VERIFIES THAT EACH BIT IS INDEPENDENT.

PROBABLE FAULT:

1. IF MODULE

CONTROL STATUS #1 TRANSFER TEST

PURPOSE:

TO VERIFY THAT BITS 01 THROUGH 05 OF CONTROL STATUS REGISTER 1 ARE NOT STUCK AT ONE OR ZERO, AND THAT THERE IS NOT BIT INTERFERENCE.

PROCEDURE:

THIS TEST WRITES ONES IN CONTROL STATUS REGISTER 1, RMCS1, THEN WRITES ZEROS AND VERIFIES THAT THE BITS ARE NOT STUCK AT ONE. THE GO BIT IS NOT TESTED IN THIS TEST.

NEXT, THE TEST CLEARS THE CONTROL STATUS REGISTER, RMCS1, WRITES ONES IN BITS 01 THROUGH 05 AND VERIFIES THAT THE BITS ARE NOT STUCK AT ZERO. THE GO BIT IS NOT TESTED.

THE TEST TRANSFERS A SHIFTING ONE BIT DATA PATTERN TO AND FROM RMCS1 AND CHECKS FOR ADJACENT BIT INTERFERENCE.

PROBABLE FAULT:

1. IF MODULE

ERROR REGISTER #1 TRANSFER TEST

PURPOSE:

TO VERIFY THAT ERROR REGISTER 1 IS NOT STUCK AT ONE OR ZERO, AND THAT THERE IS NOT BIT INTERFERENCE.

PROCEDURE:

THIS TEST WRITES ONES IN ERROR REGISTER 1, RMER1, THEN WRITES ZEROS AND VERIFIES THAT THE REGISTER IS NOT STUCK AT ONE.

```
                        "UNSAFE" IS NOT TESTED DURING THIS TEST.  IN ORDER TO  LIMIT  THE
                        PROBABLE  FAULTS TO ONE OR TWO MODULES, THE TEST IS EXECUTED IN 3
```

PARTS WITH EACH PART TESTING THOSE BITS WHOSE PRESET FUNCTIONS ARE DERIVED FROM THE SAME MODULE.

THE TEST WRITES ZEROS IN ERROR REGISTER 1, RMER1, THEN WRITES ONES AND VERIFIES THAT THE REGISTER IS NOT STUCK AT ZERO.

FINALLY, THE TEST WRITES A SHIFTING ONE BIT PATTERN IN RMER1 AND CHECKS FOR ADJACENT BIT INTERFERENCE.

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

3. DS MODULE

ERROR REGISTER #2 TRANSFER TEST

PURPOSE:

TO VERIFY THAT ERROR REGISTER 2, RMER2, IS NOT STUCK AT ONE, STUCK AT ZERO, AND THAT THERE IS NOT BIT INTERFERENCE.

PROCEDURE:

THE TEST WRITES ONES THEN WRITES ZEROS IN RMER2 AND VERIFIES THAT NONE OF THE BITS ARE STUCK AT ONE. "SKI" AND "DVC" ARE NOT TESTED. IN ORDER TO LIMIT THE NUMBER OF PROBABLE FAULTS TO ONE OR TWO MODULES, THE TEST IS EXECUTED IN 3 PARTS WITH EACH PART TESTING THOSE BITS WHOSE PRESET FUNCTIONS ARE DERIVED FROM THE SAME MODULE.

THEN THE TEST WRITES ZEROS IN ERROR REGISTER 2, AND WRITES ONES VERIFYING THAT THE REGISTER IS NOT STUCK AT ZERO.

FINALLY, THE TEST WRITES A SHIFTING ONE BIT PATTERN IN THE REGISTER AND VERIFIES THAT ALL BIT POSITIONS ARE INDEPENDENT.

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

3. DS MODULE

CLEAR OFFSET STUCK ACTIVE TEST

PURPOSE:

TO VERIFY THAT THE SIGNAL WHICH CLEARS OFFSET MODE IS NOT STUCK IN ACTIVE STATE.

PROCEDURE:

THE TEST WRITES A ONE IN THE OFFSET DIRECTION BIT WHICH IS CLEARED BY THE SIGNAL AND VERIFIES THAT A ONE CAN BE READ BACK.

PROBABLE FAULT:

1. IF MODULE

2. DS MODULE

OFFSET REGISTER TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE OFFSET REGISTER IS NOT STUCK AT ONE, STUCK AT ZERO, AND THAT THERE IS NO ADJACENT BIT INTERFERENCE.

PROCEDURE:

THE OFFSET REGISTER, RMOF, IS WRITTEN WITH ONES, THEN WRITTEN WITH ZEROS AND READ TO VERIFY THAT NONE OF THE BITS ARE STUCK AT ONE.

THEN THE OFFSET REGISTER IS WRITTEN WITH ZEROS AND WRITTEN WITH ONES TO VERIFY THAT THE REGISTER IS NOT STUCK AT ZERO.

FINALLY, THE OFFSET REGISTER IS TESTED WITH A SHIFTING ONE BIT PATTERN.

PROBABLE FAULT:

1. IF MODULE

SERIAL NUMBER TEST

PURPOSE:

TO VERIFY THAT THE SERIAL NUMBER CAN BE READ.

PROCEDURE:

THE TEST READS THE SERIAL NUMBER REGISTER SEVERAL TIMES AND VERIFIES THAT THE NUMBER IS THE SAME EACH TIME.

PROBABLE FAULT:

1. CS MODULE

CONTROL BUS PARITY DETECTION TEST

PURPOSE:

TO TEST THE RM03'S PARITY CHECKING LOGIC FOR THE MASSBUS ASYNCHRONOUS CONTROL BUS.

PROCEDURE:

THIS TEST WRITES A SHIFTING ONE BIT DATA PATTERN IN THE DISK ADDRESS REGISTER USING "PAT" TO CONTROL THE STATE OF THE PARITY BIT. "PAR" STATUS, BIT 03 OF RMER1, IS CHECKED AFTER EACH PATTERN IS TRANSFERRED..NOTE THE FOLLOWING TABLE SHOWS A SET OF TEST PATTERNS THAT COULD BE USED INSTEAD OF A SHIFTING ONE BIT PATTERN.

DATA PATTERN	PAT	PAR
000000	1	1
075266	0	0
163753	0	0
116535	1	1

PROBABLE FAULT:

1. IF MODULE

2. ASSYNCHRONOUS MASSBUS MODULE

CONTROL BUS PARITY GENERATION TEST

PURPOSE:

TO TEST THE RM03'S PARITY GENERATING LOGIC FOR THE MASSBUS ASYNCHRONOUS CONTROL BUS.

PROCEDURE:

THE TEST TRANSFERS A SHIFTING ONE BIT DATA PATTERN TO THE DISK ADDRESS REGISTER. AFTER EACH PATTERN IS READ BACK, "MASSBUS CONTROL BUS PARITY ERROR" IS TESTED AND SHOULD BE ZERO..NOTE THE FOLLOWING SET OF TEST PATTERNS COULD BE USED INSTEAD OF THE SHIFTING ONE BIT PATTERN.

DATA PATTERN	MCPE
000000	0
056747	0
135672	0
163135	0

PROBABLE FAULT:

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

RMDA, RMDC FAULT TEST

PURPOSE:

TO VERIFY THAT THERE ARE NOT FAULTS WHICH INHIBIT THE PROGRAM FROM WRITING RMDC AND RMDA. SPECIFICALLY, THESE FAULTS INCLUDE:

."GO H" STUCK HIGH, WHICH WOULD INHIBIT THE REGISTER LOAD FUNCTION,

."RIP L" STUCK LOW, WHICH WOULD CONSTANTLY CLEAR THE REGISTER

."EBL" STUCK, WHICH WOULD INHIBIT THE CLOCK FUNCTION.

PROCEDURE:

THE TEST WRITES AND READS BOTH RMDC, AND RMDA. WITH ZEROS, THEN ONES. THE TEST PASSES IF EITHER REGISTER CAN BE WRITTEN

```
896                                         WITH ONES.
```

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

3. CS MODULE

DISK ADDRESS TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE DISK ADDRESS REGISTER IS NOT STUCK AT ONE OR ZERO, AND THAT THERE IS NOT BIT INTERFERENCE.

PROCEDURE:

THIS TEST PRESETS THE DISK ADDRESS TO A NONZERO VALUE, THEN USES A MOVE TO CLEAR THE REGISTER.THE TEST THEN READS RMDA AND VERIFIES THAT NONE OF THE BITS ARE STUCK AT ONE.

THEN THE TEST PRECLEARS THE MASSBUS ADAPTER DISK ADDRESS REGISTER (RMDA), LOADS IT TO ALL ONES, AND VERIFIES THAT NONE OF THE BITS ARE STUCK AT ZERO.

A SHIFTING ONE BIT PATTERN IS TRANSFERRED TO AND FROM THE DISK ADDRESS REGISTER, RMDA, AND THE TEST VERIFIES THAT EACH BIT IS INDEPENDENT.

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

DESIRED CYLINDER TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE DESIRED CYLINDER ADDRESS REGISTER, RMDC, IS NOT STUCK AT ONE OR ZERO, AND THAT THERE IS NOT BIT INTERFERENCE.

953 PROCEDURE:

THIS TEST WRITES ONES IN THE DESIRED CYLINDER REGISTER RMDC, THEN WRITES ZEROS AND VERIFIES THAT THE REGISTER IS NOT STUCK AT ONE.

THEN THE TEST WRITES ZEROS IN THE DESIRED CYLINDER REGISTER, RMDC, WRITES ONES AND VERIFIES THAT THE REGISTER IS NOT STUCK AT ZERO.

FINALLY, A SHIFTING 1 BIT PATTERN IS TRANSFERRED TO AND FROM RMDC AND THE PROGRAM CHECKS FOR BIT INTERFERENCE.

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

ILLEGAL REGISTER TEST

PURPOSE:

TO TEST ILLEGAL REGISTER ERROR DETECTION IN THE RM03.

PROCEDURE:

THIS TEST READS ALL LEGAL REGISTERS AND VERIFIES THAT "ILR", BIT 2 OF RMER1 DOES NOT SET. THEN, TO THE EXTENT ALLOWED BY THE MASSBUS CONTROLLER, IT READS ILLEGAL REGISTERS AND VERIFIES THAT "ILR" IS SET.

PROBABLE FAULT:

1. IF MODULE

2. ASSYNCHRONOUS MASSBUS MODULE

RESET GO BY INIT TEST

PURPOSE:

TO VERIFY THAT GO CAN BE RESET BY INITIALIZE.

PROCEDURE:

THE TEST SETS GO THEN CLEARS GO USING MASSBUS INITIALIZE, I.E., CONTROLLER CLEAR.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

DIAGNOSTIC MODE TEST

PURPOSE:

TO VERIFY THAT "DIAGNOSTIC MODE", BIT 0 OF RMMR1, IS NOT STUCK AT ONE OR ZERO.

PROCEDURE:

THE RM03 IS INITIALIZED AND "DMD" IS CHECKED FOR ZERO. "DMD" IS WRITTEN WITH ONE AND READ TO VERIFY THAT IT IS NOT STUCK AT ZERO, THEN WRITTEN WITH ZERO AND READ TO VERIFY THAT IT IS NOT STUCK AT ONE.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

MOL TEST

PURPOSE:

TO VERIFY THAT "MEDIUM ON LINE" STATUS CAN BE SET AND RESET USING MAINTENANCE UNIT READY.

```
                                        PROCEDURE:
```

AFTER INITIALIZING THE SUBSYSTEM, THE TEST SETS "DIAGNOSTIC MODE: AND READS THE DRIVE STATUS REGISTER, RMDS, EXPECTING MOL, BIT 12 TO BE ZERO. "MAINTENANCE UNIT READY", BIT 9 OF RMMR1, IS SET AND MOL SHOULD BE ONE. THE TEST THEN WRITES A ZERO IN MUR AND READS RMDS, VERIFYING THAT "MEDIUM ON LINE" IS ZERO.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

WRITE LOCK TEST

PURPOSE:

TO VERIFY THAT "WRITE LOCK" STATUS, WRL, CAN BE SET AND RESET USING "MAINTENANCE WRITE PROTECT", MWP.

PROCEDURE:

WITH DIAGNOSTIC MODE SET, THE PROGRAM SETS MWP, BIT 03 OF RMMR1, AND READS RMDS TO VERIFY THAT WRL, BIT 11 IS SET. THEN MWP IS RESET AND WRL SHOULD BE ZERO.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

DRIVE FAULT TEST

PURPOSE:

TO VERIFY THAT "DEVICE CHECK", DVC, AND "UNSAFE", UNS, CAN BE SET AND RESET USING "MAINTENANCE DRIVE FAULT", MDF.

PROCEDURE:

WITH DIAGNOSTIC MODE SET, THE PROGRAM SETS MDF, BIT 06 OF RMMR1, AND READS RMER3 TO VERIFY THAT DVC, BIT 07 IS SET RMER1 IS ALSO READ AND UNS, BIT 14 SHOULD ALSO BE SET. THEN MDF IS RESET

```
1117                                    AND DVC AND UNS SHOULD BE RESET.
```

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

SEEK ERROR TEST

PURPOSE:

TO VERIFY THAT "SEEK ERROR", SKI, CAN BE SET AND RESET USING "MAINTENANCE SEEK ERROR", MSER.

PROCEDURE:

WITH DIAGNOSTIC MODE SET, THE TEST SETS MSER, BIT 07 OF RMMR1 AND READS RMER3 TO VERIFY THAT SKI, BIT 14 IS SET. MSER IS RESET AND SKI SHOULD RESET.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

PIP TEST

PURPOSE:

TO VERIFY THAT "POSITIONING IN PROGRESS", PIP, CAN BE SET AND RESET USING "MAINTENANCE ON CYLINDER", MOC.

PROCEDURE:

DIAGNOSTIC MODE IS SET THEN MOC, BIT 08 OF RMMR1 IS SET AND PIP, BIT 13 OF RMDS, SHOULD BE ZERO. MOC IS THEN RESET AND PIP SHOULD BE ONE.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

1174

EBL TEST

PURPOSE:

TO VERIFY THAT END OF BLOCK STATUS "EBL" CAN BE SET AND RESET USING DIAGNOSTIC END OF BLOCK "DEBL".

PROCEDURE:

THE PROGRAM SETS DIAGNOSTIC MODE AND VERIFIES THAT EBL IS RESET. THEN IT SETS DEBL AND VERIFIES THAT EBL IS SET. FINALLY, THE TEST TRANSFERS A SHIFTING ONE BIT TO RMMR1, AND CHECKS FOR DEBL BEING SET BY AN ADJACENT BIT.

PROBABLE FAULT:

1. CS MODULE

LAST SECTOR, LAST TRACK TEST

PURPOSE:

TO VERIFY THE DESIRED TRACK/SECTOR PLA ON THE DS MODULE USING RMMR1, BITS 01 AND 02.

PROCEDURE:

THE TEST WRITES ALL POSSIBLE PATTERNS IN THE DISK ADDRESS REGISTER, RMDA, AND VERIFIES "LS" AND "LS/T" STATUS FOR EACH PATTERN. THE PROCEDURE IS DONE ONCE FOR 16 BIT FORMAT AND ONCE FOR 18 BIT FORMAT.

PROBABLE FAULT:

1. DS MODULE

2. CS MODULE

RMDA COUNT TEST

PURPOSE

TO VERIFY THAT THE DISK ADDRESS REGISTER (RMDA) INCREMENTS

1231 PROPERLY.

PROCEDURE:

THE TEST INCREMENTS RMDA USING DIAGNOSTIC END OF BLOCK "DEBL" AND VERIFIES THE RESULT IN BOTH 16 AND 18 BIT FORMAT.

PROBABLE FAULT:

1. DS MODULE

RMDC COUNT TEST

PURPOSE:

TO VERIFY THAT THE DESIRED CYLINDER REGISTER, RMDC, INCREMENTS PROPERLY.

PROCEDURE:

THE PROGRAM INCREMENTS RMDC USING DIAGNOSTIC END OF BLOCK, "DEBL", VERIFYING THAT RMDC INCREMENTS THROUGH A COMPLETE CYCLE.

PROBABLE FAULT:

1. DS MODULE

LBT TEST

PURPOSE:

TO INSURE THAT LAST BLOCK TAKEN STATUS, "LBT", CLEARS WHEN RMDA IS WRITTEN, AND SETS WHEN THE LAST SECTOR IS TRANSFERRED.

PROCEDURE:

THE TEST USES DIAGNOSTIC EBL TO SET LBT, AND TRANSFERS TO RMDA TO RESET LBT.

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

COMPOSITE ERROR TEST

PURPOSE:

TO TEST "COMPOSITE ERROR", BIT 14 OF RMDS.

PROCEDURE:

THE TEST USES INITIALIZE AND DIAGNOSTIC MODE TO FORCE ALL ERRORS TO ZERO THEN VERIFIES THAT "ERR" IS ZERO. EACH ERROR IS INDIVIDUALLY SET AND "ERR" SHOULD BE ONE FOR EVERY ERROR TESTED. ADDRESSES #2 AND #17 OF THE COMPOSITE ERROR PLA ARE NOT TESTED. "ABORT" AND "EXCEPTION" OUTPUTS OF THE PLA ARE NOT TESTED. THE TEST FAILS IF ERR IS NOT ZERO WITH ALL SET ARGUMENTS ZERO OR IF ERR IS NOT ONE WITH ANY SET ARGUMENT ONE.

PROBABLE FAULT:

1. IF MODULE

WRITE GO TEST

PURPOSE:

TO VERIFY THAT GO CAN BE SET.

PROCEDURE:

THE TEST ENABLES THE DEBUG CLOCK, THEN TRANSFERS A NOP FUNCTION CODE AND GO BIT TO RMCS1, VERIFYING THAT GO SETS. ALL FUNCTION CODES ARE TESTED.

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

BRANCH MULTIPLEXOR TEST

PURPOSE:

TO VERIFY THAT THE OUTPUT OF THE COMMAND SEQUENCER BRANCH

1344 MULTIPLEXOR DOES NOT HAVE A FAULT.

PROCEDURE:

WITH DEBUG CLOCK ENABLED, THE TEST USES VARIOUS FUNCTION CODES AND REGISTER CONDITIONS TO ADDRESS THE TEST BIT MULTIPLEXOR SUCH THAT THE TEST BIT, BIT12 OF RMMR2, CAN BE CHECKED FOR A STUCK FAULT.

PROBABLE FAULT:

1. CS MODULE

SET/RESET GO TEST

PURPOSE:

TO VERIFY THAT GO CAN BE SET AND RESET.

PROCEDURE:

THE SUBSYSTEM IS INITIALIZED AND PUT IN DIAGNOSTIC MODE WITH "DEBUG CLOCK ENABLE", BIT 14 OF RMMR1 SET.CERTAIN FUNCTION CODES ARE WRITTEN IN RMCS1 AND THE PROGRAM READS RMCS1 TO VERIFY THAT GO IS SET. RMDS IS ALSO READ TO VERIFY THAT "DRY" IS RESET. THEN THE PROGRAM STEPS THE DEBUG CLOCK USING BIT 15 OF RMMR1 AND VERIFIES THAT "GO" RESETS AND "DRY" SETS. USING A FUNCTION CODE THAT RESETS GO AT A DIFFERENT PROM ADDRESS. THE TEST FAILS IF GO DOES NOT SET OR CANNOT BE RESET BY THE COMMAND SEQUENCER. THE TEST ALSO FAILS IF "DRIVE READY" IS NOT THE COMPLIMENT OF GO.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

END 1 RESET GO TEST

PURPOSE:

TO VERIFY THAT THE COMMAND SEQUENCER CAN RESET GO AT THE END1 LOCATION.

PROCEDURE:

THE TEST EXECUTES RELEASE, SEARCH AND ILLEGAL FUNCTION CODE 32 IN DIAGNOSTIC MODE AND VERIFIES THAT GO RESETS ON THE SPECIFIED CLOCK CYCLE.

PROBABLE FAULT:

1. CS MODULE

SET PULSE TEST

PURPOSE:

TO VERIFY THAT THE COMMAND SEQUENCER CAN GENERATE SET PULSE.

PROCEDURE:

WHICH DEBUG CLOCK ENABLED, THE TEST STEPS THE COMMAND SEQUENCER THROUGH PARTS OF VARIOUS FUNCTION CODES AND CHECKS CONTINUE, BIT 06 OF RMMR1 TO DETERMINE IF SET PULSE IS BEING GENERATED.

PROBABLE FAULT:

1. CS MODULE

SET/RESET IVC TEST

PURPOSE:

TO TEST "INVALID COMMAND" STATUS FOR EACH FUNCTION CODE.

PROCEDURE:

THE PROGRAM RESETS VOLUME VALID USING "MAINTENANCE UNIT READY", BIT09 OF RMMR1, THEN LOADS THE FUNCTION CODE AND GO IN RMCS1. EACH FUNCTION CODE IS TESTED AND "IVC", BIT 12 OF RMER2 IS CHECKED.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

1457

SET LSC TEST

PURPOSE:

TO VERIFY THAT "LOSS OF SYSTEM CLOCK" CAN SET AND RESET.

PROCEDURE:

THE TEST ENABLES THE DEBUG CLOCK AND SETS THE GO BIT. AFTER WAITING ENOUGH TIME FOR THE ONE SHOT TO SET, THE TEST DISABLES THE DEBUG CLOCK AND VERIFIES THAT LSC SETS.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

DECODE TEST

PURPOSE:

TO VERIFY THAT THE "DECODE" FLOP ON THE IF MODULE SETS WITH THE LEADING EDGE OF "SET PULSE" EXCEPT WHEN "COMPOSITE ERROR" IS ACTIVE.

PROCEDURE:

THE TEST USES "VOLUME VALID" AND "OCCUPPIED" TO DETERMINE IF THE DECODE FLOP IS SET OR RESET. INITIALLY, VV AND OCCUPPIED ARE RESET AND THE TEST EXECUTES THOSE COMMANDS WHICH SET VV OR OCC AND VERIFIES THAT ONE OR BOTH BITS SET. THE SAME COMMANDS ARE EXECUTED AGAIN WITH COMPOSITE ERROR SET, AND THE TEST VERIFIES THAT NEITHER BIT SETS.

PROBABLE FAULT:

1. IF MODULE

SET/RESET VOLUME VALID TEST

PURPOSE:

TO VERIFY THAT "VOLUME VALID" RESETS WITH THE LEADING EDGE OF UNIT READY, AND SETS WITH PACK ACKNOWLEDGE AND READ IN PRESET COMMANDS.

PROCEDURE:

USING "MAINTENANCE UNIT READY", BIT 9 OF RMMR1, THIS TEST FORCES A ZERO TO ONE TRANSITION OF UNIT READY AND VERIFIES THAT VOLUME VALID,BIT 6 OF RMDS IS ZERO. THEN THE TEST EXECUTES A PACK ACKNOWLEDGE COMMAND, VERIFYING THAT VV SETS. THE PROCEDURE IS REPEATED WITH A READ IN PRESET COMMAND.

PROBABLE FAULT:

1. IF MODULE

ILLEGAL FUNCTION TEST

PURPOSE:

TO TEST ILLEGAL FUNCTION ERROR IN THE RM03.

PROCEDURE:

WITH DIAGNOSTIC CLOCK ENABLED TO INHIBIT THE COMMAND SEQUENCER, THIS TEST VERIFIES THAT "ILF", BIT 0 OF RMER1, IS OFF FOR LEGAL FUNCTION CODES AND ON FOR ILLEGAL FUNCTIONCODES. THE STATUS OF THE "GO" BIT IS IGNORED.

PROBABLE FAULT:

1. IF MODULE

OCCUPIED TEST

PURPOSE:

TO VERIFY THAT "OCCUPIED" IS SET DURING DATA TRANSFERS AND IS RESET FOR ALL OTHER COMMANDS.

PROCEDURE:

FOR EACH DATA TRANSFER COMMAND, "OCC", BIT 15 OF RMMR1

```
                                        SHOULD BE ONE. DEBUG CLOCK IS ENABLED TO PREVENT GO FROM
                                        RESETTING BEFORE STATUS IS SAMPLED.
```

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

READ IN PRESET TEST

PURPOSE:

TO VERIFY THAT "READ IN PRESET" COMMAND IS DECODED, AND IN PARTICULAR, TO VERIFY THAT "IFS READ IN CMD L" IS NOT STUCK AT ONE.

PROCEDURE:

EACH VISIBLE STATUS OR REGISTER BIT WHICH IS CLEARED BY "READ IN PRESET" IS SET. THEN THE RIP COMMAND IS EXECUTED AND THE TEST VERIFIES THAT ONE OR MORE BITS ARE CLEARED. THE FOLLOWING ARE USED DURING THE TEST.

. BITS 10-12 OF RMOF ARE SET BY A MOVE INSTRUCTION AND THE TEST PASSES IF THESE BITS ARE ZERO AFTER THE RIP COMMAND.

. THE DESIRED CYLINDER REGISTER, RMDC, IS SET WITH A MOVE INSTRUCTION AND THE TEST PASSES IF BITS 00-03, OR BITS 04-07, OR BITS 08-09 ARE ZERO AFTER THE RIP COMMAND.

. THE DISK ADDRESS REGISTER, RMDA, IS SET WITH A MOVE INSTRUCTION AND THE TEST PASSES IF BITS 00-03, OR BITS 04-07, OR BITS 08-11, OR BITS 12-15 ARE ZERO AFTER THE RIP COMMAND.

THE TEST FAILS IF NONE OF THE PRESET TERMS ARE ZERO AFTER THE RIP COMMAND.

PROBABLE FAULT:

1. IF MODULE

RIP/RMOF TEST

PURPOSE:

TO VERIFY THAT "READ IN PRESET" RESETS FMT16, ECI AND HCI, BITS 10,11 AND 12 OF RMOF.

PROCEDURE:

FMT16, ECI AND HCI ARE SET, THEN A RIP COMMAND IS EXECUTED AND EACH BIT SHOULD BE ZERO.

PROBABLE FAULT:

1. IF MODULE

RMDA/RMDC/RIP TEST

PURPOSE:

TO VERIFY THAT "READ IN PRESET" RESETS THE DESIRED CYLINDER ADDRESS, RMDC, AND THE DISK ADDRESS, RMDA.

PROCEDURE:

RMDA AND RMDC ARE PRESET THEN TESTED FOR ZERO AFTER THE RIP COMMAND.

PROBABLE FAULT:

1. DS MODULE

OFFSET COMMAND TEST

PURPOSE:

TO VERIFY THAT "OFFSET MODE" SETS WITH OFFSET COMMAND.

PROCEDURE:

THE TEST EXECUTES OFFSET COMMAND AND VERIFIES THAT "OM", BIT 00 OF RMDS IS ONE.

PROBABLE FAULT:

1. IF MODULE

RETURN TO CENTER TEST

PURPOSE:

TO VERIFY THAT "RETURN TO CENTER" RESETS OFFSET MODE.

PROCEDURE:

OFFSET MODE, BIT 00 OF RMDS, IS SET WITH OFFSET COMMAND, THEN THE TEST EXECUTES A RETURN TO CENTER COMMAND AND VERIFIES THAT OFFSET MODE RESETS. OFFSET DIRECTION IS ALSO SET AND CHECKED FOR ZERO AFTER THE COMMAND.

PROBABLE FAULT:

1. IF MODULE

RMDC CLEAR OFFSET TEST

PURPOSE:

TO VERIFY THAT CLEAR OFFSET IS ACTIVE WHEN THE DESIRED CYLINDER ADDRESS IS WRITTEN.

PROCEDURE:

THE TEST EXECUTES AN OFFSET COMMAND, WRITES RMDC, AND VERIFIES THAT OM, BIT 00 OF RMDS IS ZERO.

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

EBL CLEAR OFFSET TEST

PURPOSE:

TO VERIFY THAT OFFSET MODE CLEARS WHEN HEAD SWITCHING OCCURS.

PROCEDURE:

THE TEST EXECUTES AN OFFSET COMMAND TO SET OFFSET MODE. AFTER SETTING THE FORMAT BIT AND LOADING THE LAST SECTOR/TRACK ADDRESS IN RMDA, THE TEST FORCES AN EBL AND VERIFIES THAT OFFSET MODE RESETS.

PROBABLE FAULT:

1. DS MODULE

RUN AND GO TEST

PURPOSE:

TO VERIFY THAT "RUN AND GO" FLOP SETS DURING READ AND WRITE COMMANDS.

PROCEDURE:

THE RM03 IS INITIALIZED AND A DATA TRANSFER COMMAND WITH GO SET IS WRITTEN IN RMCS1. "RUN AND GO", BIT 14 OF RMMR1 SHOULD BE ONE FOR EACH DATA COMMAND. THE DEBUG CLOCK IS ENABLED SO THAT GO DOES NOT RESET BEFORE STATUS IS TESTED.

PROBABLE FAULT:

1. CS MODULE

2. SYNCHRONOUS MASSBUS MODULE

SET IAE TEST

PURPOSE:

TO VERIFY THAT INVALID ADDRESS ERROR CAN SET.

PROCEDURE:

THE TEST LOADS INVALID SECTOR, TRACK AND CYLINDER ADDRESSES AND EXECUTES A SEARCH COMMAND, VERIFYING THAT "IAE" SETS. THE PROCESS IS REPEATED WITH A DIFFERENT COMMAND IF THE IAE DOES NOT SET, AND THE TEST FAILS IF IAE CANNOT BE SET.

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

SEARCH, SEEK, READ, WRITE TEST

PURPOSE:

TO VERIFY THAT THE "SCH SK R OR W" DECODE ON THE IF MODULE IS CORRECT FOR ALL FUNCTION CODES.

PROCEDURE:

THE TEST LOADS INVALID SECTOR, TRACK AND CYLINDER ADDRESSES AND EXECUTES EACH COMMAND TO WHERE SET PULSE IS ACTIVE AND VERIFIES THE DECODE BY CHECKING "IAE".

PROBABLE FAULT:

1. IF MODULE

INVALID SECTOR/TRACK TEST

PUPOSE:

TO VERIFY THAT INVALID SECTOR AND TRACK ADDRESSES ARE DETECTED.

PROCEDURE:

THE TEST LOADS THE TEST PATTERN IN RMDA AND EXECUTES A SEARCH COMMAND, VERIFYING THAT "IAE" SETS.

PROBABLE FAULT:

1. DS MODULE

2. TRACK ADDRESS OPTION JUMPER

INVALID CYLINDER TEST

PURPOSE:

TO VERIFY THAT INVALID CYLINDER ADDRESSES ARE DETECTED.

PROCEDURE:

THE TEST LOADS THE TEST PATTERN IN RMDC AND EXECUTES A SEARCH COMMAND, VERIFYING THAT "IAE" SETS.

PROBABLE FAULTS:

1. DS MODULE

2. CYLINDER ADDRESS OPTION JUMPER

SET AOE TEST

PRUPOSE:

TO VERIFY THAT ADDRESS OVERFLOW ERROR IS DETECTED.

PROCEDURE:

THE TEST LOADS THE ADDRESS OF THE LAST SECTOR IN RMDA AND RMDC, THEN INITIATES A DATA COMMAND WITH DEBUG CLOCK ENABLED. END OF BLOCK IS FORCED TO INCREMENT THE SECTOR ADDRESS, AND THE TEST VERIFIES THAT "AOE" IS SET.

PROBABLE FAULT:

1. DS MODULE

SET RMR TEST

PURPOSE:

TO VERIFY THAT "REGISTER MODIFICATION REFUSED" SETS WHEN A REGISTER IS WRITTEN WHILE GO IS SET, EXCEPT WHEN THE ATTENTION OR MAINTENANCE REGISTER IS WRITTEN.

PROCEDURE:

"DEBUG CLOCK ENABLE" IS SET TO INHIBIT THE COMMAND SEQUENCER, THEN A NOP COMMAND AND GO BIT IS WRITTEN IN RMCS1. WITHOUT STEPPING THE DEBUG CLOCK, THE TEST WRITES RMMR AND RMAS, WHICH SHOULD NOT SET RMR STATUS. THEN RMDA IS WRITTEN AND RMR STATUS, BIT 02 OF RMER1, SHOULD BE ONE.

PROBABLE FAULT:

1. IF MODULE

PGM STATUS CHECK

PURPOSE:

TO VERIFY THAT THE PROGRAMMABLE STATUS BIT AND THE DRIVE REQUEST STATUS BIT ARE COMPATABLE.

PROCEDURE:

THE TEST REPORTS AN ERROR IF PGM IS ON AND DRQ IS OFF. PGM IS NOT PREDICTABLE IN THE CASE WHERE DRQ IS ON BECAUSE OF THE PORT SELECT SWITCH.

PROBABLE FAULT:

1. IF MODULE

DVA/DPR STATUS CHECK PURPOSE:

TO VERIFY THAT DRIVE PRESENT STATUS AND DEVICE AVAILABLE STATUS ARE SET.

PROCEDURE:

DPR AND DVA ARE TESTED AND BOTH SHOULD BE ON.

PROBABLE FAULT:

1. IF MODULE

PORT REQUEST TEST, PART 1/3

PURPOSE:

TO VERIFY THAT THE PORT REQUEST FLOPS ON THE IF MODULE SET WHEN THE PROGRAM READS RMCS1.

PROCEDURE:

THE TEST EXECUTES A RELEASE COMMAND, THEN, ASSUMING THE PORT IS RELEASED, IT READS RMCS1, THEN READS RMMR2 AND VERIFIES THAT ONE OF THE PORT REQUEST FLOPS IS SET.

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

PORT REQUEST TEST, PART 2/3

PURPOSE:

TO VERIFY THAT THE PORT REQUEST FLOPS ON THE IF MODULE SET WHEN THE PROGRAM WRITES RMAS.

PROCEDURE:

THE TEST EXECUTES A RELEASE COMMAND THEN WRITES RMAS AND READS RMMR2, VERIFYING THAT ONE OF THE REQUEST FLOPS IS SET.

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

DATA COMMAND TESTS

PURPOSE:

TO VERIFY THE COMMAND SEQUENCER DURING DATA COMMANDS.

2011 PROCEDURE:

THIS TEST, LIKE RECALIBRATE, SEEK, AND SEARCH TESTS, USES THE MAINTENANCE REGISTER TO SIMULATE DRIVE CONDITIONS AND FORCE THE COMMAND SEQUENCER THROUGH EACH BRANCH PATH. ADDITIONAL ITEMS WHICH ARE TESTED INCLUDE OFFSET PLUS AND MINUS ON THE TAG BUS AND "ENABLE SEARCH", BIT 11 OF RMMR1.

PROBABLE FAULT:

1. CS MODULE

DATA SEQUENCER READ TEST

PURPOSE:

TO TEST THE CS AND DS MODULES FOR EXECUTION OF A READ COMMAND.

PROCEDURE:

THE RM03 IS INITIALIZED AND, USING DIAGNOSTIC MODE, A READ COMMAND IS STEPPED TO THE POINT WHERE THE COMMAND SEQUENCER ENABLES SEARCH. THE PROGRAM THEN STARTS STEPPING THE DATA SEQUENCER ON THE CS MODULE AND MONITORS HARDWARE OPERATION USING THE FOLLOWING STATUS:

.PLFS, BIT 10 OF RMMR1, SHOWS THE PERIOD WHEN THE DATA PROM HAS ENABLED SYNC BYTE DETECTION;

.PDA, BIT 08 OF RMMR1, SHOWS WHEN THE DATA PROM IS IN THE DATA AREA OF THE SECTOR;

.PHA, BIT 07 OF RMMR1, SHOWS WHEN THE DATA PROM IS IN THE HEADER AREA OF THE SECTOR;

.BB00, BIT00 OF RMMR2, SHOWS THE STATE OF WRITE GATE PROVIDING TAG IS HIGH;

.BB01, BIT 01 OF RMMR2, SHOWS THE STATE OF READ GATE PROVING TAG IS HIGH;

.WC, BIT 05 OF RMMR1, SHOWS THE STATE OF WORD CLOCK;

.EECC, BIT 04 OF RMMR1, SHOWS WHEN ECC CHARACTER IS ENABLED DURING WRITE;

.ECRC, BIT 09 OF RMMR1, SHOWS WHEN CRC CHARACTER IS ENABLED DURING WRITE;

.WD, BIT 03 OF RMMR1, SHOWS THE STATE OF WRITE DATA.

ADDITIONALLY, STATUS BITS IN RMER1 ARE USED TO TEST COMMAND EXECUTION. THE WRITE ONLY BITS OF RMMR1 ARE USED TO SIMULATE DRIVE STATUS AND CONTROL SIGNALS, AND TO SPECIFY READ DATA.

PROBABLE FAULT:

1. CS MODULE

2. DS MODULE

DATA SEQUENCER WRITE TEST

PURPOSE:

TO TEST THE CS AND DS MODULES FOR EXECUTION OF A WRITE COMMAND.

PROCEDURE:

THE PROCEDURE IS THE SAME AS THE DATA SEQUENCER READ TEST.

PROBABLE FAULT:

1. CS MODULE

2. DS MODULE

DATA SEQUENCER FORMAT TEST

PURPOSE:

TO TEST CS AND DS MODULES FOR EXECUTION OF A FORMAT COMMAND.

PROCEDURE:

THE PROCEDURE IS THE SAME AS THE DATA SEQUENCER READ TEST.

PROBABLE FAULT:

1. CS MODULE

2. DS MODULE

18 BIT WORD CLOCK TEST

PURPOSE:

TO TEST THE GENERATION OF WORD CLOCK IN 18 BIT FORMAT.

PROCEDURE:

DIAGNOSTIC MODE IS USED TO EXECUTE A READ COMMAND. THE PROGRAM VERIFIES THAT WORD CLOCK SWITCHES TO 18 BIT FORMAT AT THE DATA AREA AND SWITCHES BACK TO 16 BIT FORMAT DURING ECC. THE TEST IS REPEATED FOR A WRITE COMMAND.

PROBABLE FAULT:

1. CS MODULE

2. DS MODULE

HCRC SET TEST

PURPOSE:

TO VERIFY THAT "HCRC" STATUS SETS WHEN THE HEADER CRC WORD IS IN ERROR.

PROCEDURE:

USING DIAGNOSTIC MODE TO EXECUTE A READ COMMAND, THE PROGRAM STEPS THE DATA SEQUENCER THROUGH THE HEADER AND PROVIDES READ DATA WHICH SHOULD CAUSE A CRC ERROR. THE TEST FAILS IF "HCRC", BIT 08 OF RMER1 DOES NOT SET.

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

HCE SET TEST

PURPOSE:

TO VERIFY THAT "HCE" STATUS SETS IF THE HEADER DOES NOT COMPARE WITH RMDC AND RMDA.

PROCEDURE:

THE PROGRAM EXECUTES A READ COMMAND USING DIAGNOSTIC MODE. READ DATA IS SUPPLIED WHICH CAUSES A HEADER COMPARE ERROR IN BITS 00-03 OF HEADER WORD ONE AND THE PROGRAM VERIFIES THAT "HCE", BIT 07 OF RMER1 IS ONE. THIS PROCESS IS REPEATED FOR BITS 04-07 AND 08-11.

PROBABLE FAULT:

1. DS MODULE

2. CS MODULE

3. IF MODULE

FER SET TEST

PURPOSE:

TO VERIFY THAT "FER" STATUS SETS WHEN THE FORMAT BIT DOES NOT COMPARE WITH THE HEADER.

PROCEDURE:

THIS PROGRAM EXECUTES A READ COMMAND USING DIAGNOSTIC MODE AND FORCES A FORMAT ERROR IN THE FIRST HEADER WORD. THE TEST FAILS IF "FER", BIT 04 OF RMER1 DOES NOT SET.

PROBABLE FAULT:

1. DS MODULE

2. CS MODULE

3. IF MODULE

DCK SET TEST

PURPOSE:

TO VERIFY THAT "DCK" STATUS SETS WHEN A CORRECTABLE ECC ERROR OCCURRS.

PROCEDURE:

THE PROGRAM SIMULATES A READ OPERATION AND FORCES AN ECC ERROR WHICH IS CORRECTABLE. "DCK", BIT 15 OF RMER1 SHOULD BE SET AND "ECH", BIT 06 OF RMER1 SHOULD BE RESET. RMEC1 AND RMEC2 SHOULD PERMIT DATA CORRECTION.

PROBABLE FAULT:

1. DS MODULE

2. CS MODULE

3. IF MODULE

ECH SET TEST

PURPOSE:

TO VERIFY THAT "ECH" STATUS SETS WHEN A NONCORRECTABLE ECC ERROR OCCURRS DURING READ.

PROCEDURE:

THIS TEST IS LIKE THE DCK SET TEST EXCEPT THE PROGRAM USES AN UNCORRECTABLE ERROR BURST.

PROBABLE FAULT:

1. DS MODULE

2. CS MODULE

3. IF MODULE

DVA/DPR

STATUS CHECK PURPOSE:

TO VERIFY THAT DRIVE PRESENT STATUS AND DEVICE AVAILABLE STATUS ARE SET.

PROCEDURE:

DPR AND DVA ARE TESTED AND BOTH SHOULD BE ON.

PROBABLE FAULT:

1. IF MODULE

PGM STATUS CHECK

PURPOSE:

TO VERIFY THAT THE PROGRAMMABLE STATUS BIT AND THE DRIVE REQUEST STATUS BIT ARE COMPATABLE.

PROCEDURE:

THE TEST REPORTS AN ERROR IF PGM IS ON AND DRQ IS OFF. PGM IS NOT PREDICTABLE IN THE CASE WHERE DRQ IS ON BECAUSE OF THE PORT SELECT SWITCH.

PROBABLE FAULT:

1. IF MODULE

SET AOE TEST

PRUPOSE:

TO VERIFY THAT ADDRESS OVERFLOW ERROR IS DETECTED.

PROCEDURE:

THE TEST LOADS THE ADDRESS OF THE LAST SECTOR IN RMDA AND RMDC, THEN INITIATES A DATA COMMAND WITH DEBUG CLOCK ENABLED.

```
2334                                    END OF BLOCK IS FORCED TO INCREMENT THE SECTOR ADDRESS,  AND  THE
2335                                    TEST VERIFIES THAT "AOE" IS SET.
```

PROBABLE FAULT:

1. DS MODULE

CLEAR OFFSET STUCK ACTIVE TEST

PURPOSE:

TO VERIFY THAT THE SIGNAL WHICH CLEARS OFFSET MODE IS NOT STUCK IN ACTIVE STATE.

PROCEDURE:

THE TEST WRITES A ONE IN THE OFFSET DIRECTION BIT WHICH IS CLEARED BY THE SIGNAL AND VERIFIES THAT A ONE CAN BE READ BACK.

PROBABLE FAULT:

1. IF MODULE

2. DS MODULE

INVALID SECTOR/TRACK TEST

PUPOSE:

TO VERIFY THAT INVALID SECTOR AND TRACK ADDRESSES ARE DETECTED.

PROCEDURE:

THE TEST LOADS THE TEST PATTERN IN RMDA AND EXECUTES A SEARCH COMMAND, VERIFYING THAT "IAE" SETS.

PROBABLE FAULT:

1. DS MODULE

2. TRACK ADDRESS OPTION JUMPER

INVALID CYLINDER TEST

PURPOSE:

TO VERIFY THAT INVALID CYLINDER ADDRESSES ARE DETECTED.

PROCEDURE:

THE TEST LOADS THE TEST PATTERN IN RMDC AND EXECUTES A SEARCH COMMAND, VERIFYING THAT "IAE" SETS.

PROBABLE FAULTS:

1. DS MODULE

2. CYLINDER ADDRESS OPTION JUMPER

SEARCH, SEEK, READ, WRITE TEST

PURPOSE:

TO VERIFY THAT THE "SCH SK R OR W" DECODE ON THE IF MODULE IS CORRECT FOR ALL FUNCTION CODES.

PROCEDURE:

THE TEST LOADS INVALID SECTOR, TRACK AND CYLINDER ADDRESSES AND EXECUTES EACH COMMAND TO WHERE SET PULSE IS ACTIVE AND VERIFIES THE DECODE BY CHECKING "IAE".

PROBABLE FAULT:

1. IF MODULE

EBL CLEAR OFFSET TEST

PURPOSE:

TO VERIFY THAT OFFSET MODE CLEARS WHEN HEAD SWITCHING OCCURS.

PROCEDURE:

THE TEST EXECUTES AN OFFSET COMMAND TO SET OFFSET MODE. AFTER SETTING THE FORMAT BIT AND LOADING THE LAST SECTOR/TRACK ADDRESS IN RMDA, THE TEST FORCES AN EBL AND VERIFIES THAT OFFSET MODE RESETS.

PROBABLE FAULT:

1. DS MODULE

SET LSC TEST

PURPOSE:

TO VERIFY THAT "LOSS OF SYSTEM CLOCK" CAN SET AND RESET.

PROCEDURE:

THE TEST ENABLES THE DEBUG CLOCK AND SETS THE GO BIT. AFTER WAITING ENOUGH TIME FOR THE ONE SHOT TO SET, THE TEST DISABLES THE DEBUG CLOCK AND VERIFIES THAT LSC SETS.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

DECODE TEST

PURPOSE:

TO VERIFY THAT THE "DECODE" FLOP ON THE IF MODULE SETS WITH THE LEADING EDGE OF "SET PULSE" EXCEPT WHEN "COMPOSITE ERROR" IS ACTIVE.

PROCEDURE:

THE TEST USES "VOLUME VALID" AND "OCCUPPIED" TO DETERMINE IF THE DECODE FLOP IS SET OR RESET. INITIALLY, VV AND OCCUPPIED ARE RESET AND THE TEST EXECUTES THOSE COMMANDS WHICH SET VV OR OCC AND VERIFIES THAT ONE OR BOTH BITS SET. THE SAME COMMANDS ARE EXECUTED AGAIN WITH COMPOSITE ERROR SET, AND THE TEST VERIFIES THAT NEITHER BIT SETS.

```
2500
2501                                    PROBABLE FAULT:
```

1. IF MODULE

SET IAE TEST

PURPOSE:

TO VERIFY THAT INVALID ADDRESS ERROR CAN SET.

PROCEDURE:

THE TEST LOADS INVALID SECTOR, TRACK AND CYLINDER ADDRESSES AND EXECUTES A SEARCH COMMAND, VERIFYING THAT "IAE" SETS. THE PROCESS IS REPEATED WITH A DIFFERENT COMMAND IF THE IAE DOES NOT SET, AND THE TEST FAILS IF IAE CANNOT BE SET.

PROBABLE FAULT:

1. DS MODULE

2. IF MODULE

PORT REQUEST TEST, PART 3/3

PURPOSE:

TO VERIFY THAT PORT REQUEST SETS WHEN ANY REGISTER EXCEPT RMAS IS WRITTEN.

PROCEDURE:

THE TEST WRITES THE DISK ADDRESS REGISTER AND VERIFIES THAT THE PORT REQUEST FLOP IS ON.

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

WRITE ATA TEST

PURPOSE:

TO VERIFY THAT ATTENTION CAN BE CLEARED BY WRITING THE ATTENTION SUMMARY REGISTER.

PROCEDURE:

THE PROGRAM RESETS AND SETS UNIT READY WHICH SHOULD CAUSE AN ATTENTION, THEN WRITES THE ATTENTION SUMMARY REGISTER AND VERIFIES THAT ATTENTION IS RESET.

PROBABLE FAULT:

1. IF MODULE

2. CS MODULE

RESET ATA BY GO TEST

PURPOSE:

TO VERIFY THAT ATA RESETS WHEN GO IS ON AND COMPOSITE ERROR IS OFF.

PROCEDURE:

THE PROGRAM SETS MAINTENANCE UNIT READY WHICH SHOULD CAUSE AN ATTENTION. THEN, WITH DEBUG CLOCK ENABLED, GO IS SET, AND ATA SHOULD BE ZERO.

PROBABLE FAULT:

1. IF MODULE

UNIT READY ATA TEST

PURPOSE:

TO VERIFY THAT ONE-ZERO AND ZERO-ONE TRANSITIONS OF UNIT READY SET ATTENTION.

2610 PROCEDURE:

THE TEST USES DIAGNOSTIC MODE TO FORCE BOTH TRANSITIONS OF UNIT READY AND VERIFIES THAT ATA SETS WITH EACH TRANSITION.

PROBABLE FAULT:

1. IF MODULE

ERROR ATA TEST

PURPOSE:

TO VERIFY THAT ATTENTION SETS WHEN COMPOSITE ERROR OCCURS WHILE GO IS OFF.

PROCEDURE:

THE PROGRAM CLEARS THE DEVICE AND SETS AN ERROR, THEN VERIFIES ATA IS ON.

PROBABLE FAULT:

1. IF MODULE

REGISTER TRANSFER ATA TEST

TO VERIFY THAT ATTENTION SETS WHEN ANY REGISTER, EXCEPT FOR RMAS AND RMCS, IS WRITTEN WHILE COMP ERROR IS SET.

PROCEDURE:

THE PROGRAM FORCES AN ERROR THEN RESETS ATTENTION FROM THE ERROR. THE PROGRAM THEN WRITES RMAS AND RMCS AND VERIFIES THAT NO ATTENTION OCCURS, AND WRITES RMDC AND VERIFIES THAT ATTENTION DOES OCCUR.

PROBABLE FAULT:

1. IF MODULE

```
                        ;PROGRAM REVISION #001

                        .TITLE  MD-11-DZRMJA-A, RM03 DISKLESS DIAGNOSTIC
                        ;*COPYRIGHT (C) 1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY DOUG RIIKONEN
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
        000001          $TN=1
                        .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<7:0>
                        ;*        7             TN128
                        ;*        6             TN64
                        ;*        5             TN32
                        ;*        4             TN16
                        ;*        3             TN8
                        ;*        2             TN4
                        ;*        1             TN2
                        ;*        0             TN1
                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
        001100          $TACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

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

                        ;*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
000006        SP=     %6              ;;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
000100        BIT06=  100
```

```
        000040          BIT05=  40
        000020          BIT04=  20
        000010          BIT03=  10
        000004          BIT02=  4
        000002          BIT01=  2
        000001          BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
        000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
        000014          TBITVEC=14              ;;"T" BIT
        000014          TRTVEC= 14              ;;TRACE TRAP
        000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
        000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
        000024          PWRVEC= 24              ;;POWER FAIL
        000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
        000034          TRAPVEC=34              ;;"TRAP" TRAP
        000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
        000064          TPVEC=  64              ;;TTY PRINTER VECTOR
        000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                        .SBTTL  RM03 REGISTER BIT DEFINITIONS

                        ;RMCS1  CONTROL STATUS REGISTER

        004000          DVA     =       BIT11           ;DEVICE AVAILABLE-READ ONLY
        000040          F4      =       BIT05           ;FUNCTION CODE
        000020          F3      =       BIT04           ;FUNCTION CODE
        000010          F2      =       BIT03           ;FUNCTION CODE
        000004          F1      =       BIT02           ;FUNCTION CODE
        000002          F0      =       BIT01           ;FUNCTION CODE
        000001          GO      =       BIT00           ;GO BIT
        000077          FNCMSK  =       000077          ;FUNCTION CODE MASK

                        ;FUNCTION CODES (BITS 01-05 OF RMCS1)

        000000          NOP     =       000000          ;NOP COMMAND
        000002          ILF02   =       000002          ;ILLEGAL COMMAND
        000004          SEEK    =       000004          ;SEEK COMMAND
        000006          RECAL   =       000006          ;RECALIBRATE COMMAND
        000010          DRVCLR  =       000010          ;DRIVE CLEAR COMMAND
        000012          RLEASE  =       000012          ;RELEASE COMMAND
        000014          OFFSET  =       000014          ;OFFSET COMMAND
        000016          RTC     =       000016          ;RETURN TO CENTERLINE COMMAND
        000020          RIP     =       000020          ;READ IN PRESET COMMAND
        000022          PAKACK  =       000022          ;PACK ACKNOWLEDGE COMMAND
```

```
        000022          PACACK  =       PAKACK
        000024          ILF24   =       000024          ;ILLEGAL COMMAND
        000026          ILF26   =       000026          ;ILLEGAL COMMAND
        000030          SEARCH  =       000030          ;SEARCH COMMAND
        000030          ILF30   =       000030          ;ILLEGAL COMMAND
        000032          ILF32   =       000032          ;ILLEGAL COMMAND
        000034          ILF34   =       000034          ;ILLEGAL COMMAND
        000036          ILF36   =       000036          ;ILLEGAL COMMAND
        000040          ILF40   =       000040          ;ILLEGAL COMMAND
        000042          ILF42   =       000042          ;ILLEGAL COMMAND
        000044          ILF44   =       000044          ;ILLEGAL COMMAND
        000046          ILF46   =       000046          ;ILLEGAL COMMAND
        000050          WCD     =       000050          ;WRITE CHECK DATA COMMAND
        000052          WCH     =       000052          ;WRITE CHECK HEADER AND DATA
        000054          ILF54   =       000054          ;ILLEGAL COMMAND
        000056          ILF56   =       000056          ;ILLEGAL COMMAND
        000060          WD      =       000060          ;WRITE DATA COMMAND
        000062          WH      =       000062          ;WRITE HEADER AND DATA COMMAND
        000064          ILF64   =       000064          ;ILLEGAL COMMAND
        000066          ILF66   =       000066          ;ILLEGAL COMMAND
        000070          RD      =       000070          ;READ DATA COMMAND
        000072          RH      =       000072          ;READ HEADER AND DATA COMMAND
        000074          ILF74   =       000074          ;ILLEGAL COMMAND
        000076          ILF76   =       000076          ;ILLEGAL COMMAND

                        ;RMDA   DISK ADDRESS REGISTER

        002000          TA4     =       BIT10           ;TRACK ADDRESS 4
        001000          TA2     =       BIT09           ;TRACK ADDRESS 2
        000400          TA1     =       BIT08           ;TRACK ADDRESS 1
        000020          SA16    =       BIT04           ;SECTOR ADDRESS 16
        000010          SA8     =       BIT03           ;SECTOR ADDRESS 8
        000004          SA4     =       BIT02           ;SECTOR ADDRESS 4
        000002          SA2     =       BIT01           ;SECTOR ADDRESS 2
        000001          SA1     =       BIT00           ;SECTOR ADDRESS 1

                        ;TRACK,SECTOR MASKS

        003400          TADMSK  =       003400          ;TRACK ADDRESS MASK
        000037          SADMSK  =       000037          ;SECTOR ADDRESS MASK

                        ;RMDS   DRIVE STATUS REGISTER

        100000          ATA     =       BIT15           ;ATTENTION ACTIVE
        040000          ERR     =       BIT14           ;COMPOSITE ERROR
        020000          PIP     =       BIT13           ;POSITIONING IN PROGRESS
        010000          MOL     =       BIT12           ;MEDIUM ON LINE
        004000          WRL     =       BIT11           ;WRITE LOCK
        002000          LBT     =       BIT10           ;LAST BLOCK TRANSFERRED
        001000          PGM     =       BIT09           ;PROGRAMMABLE
        000400          DPR     =       BIT08           ;DRIVE PRESENT
        000200          DRY     =       BIT07           ;DRIVE READY
        000100          VV      =       BIT06           ;VOLUME VALID
        000001          OM      =       BIT00           ;OFFSET MODE ACTIVE

                        ;RMER1  ERROR REGISTER #1
```

```

     100000          DCK     =       BIT15           ;DATA CHECK ERROR
     040000          UNS     =       BIT14           ;DRIVE UNSAFE
     020000          OPI     =       BIT13           ;OPERATION INCOMPLETE
     010000          DTE     =       BIT12           ;DRIVE TIMING ERROR
     004000          WLE     =       BIT11           ;WRITE LOCK ERROR
     002000          IAE     =       BIT10           ;INVALID ADDRESS ERROR
     001000          AOE     =       BIT09           ;ADDRESS OVERFLOW ERROR
     000400          HCRC    =       BIT08           ;HEADER CRC ERROR
     000200          HCE     =       BIT07           ;HEADER COMPARE ERROR
     000100          ECH     =       BIT06           ;ECC "HARD" ERROR
     000040          WCF     =       BIT05           ;WRITE CLOCK FAILURE
     000020          FER     =       BIT04           ;FORMAT ERROR
     000010          PAR     =       BIT03           ;PARITY ERROR
     000004          RMR     =       BIT02           ;REGISTER MODIFICATION REFUSED
     000002          ILR     =       BIT01           ;ILLEGAL REGISTER
     000001          ILF     =       BIT00           ;ILLEGAL FUNCTION

     115760          NDTMSK  =       DCK!DTE!WLE!AOE!HCRC!HCE!ECH!WCF!FER
                     ;"NDTMSK" IS USED TO MASK ERROR REGISTER 1 DURING NON - DATA
                     ;COMMANDS, I.E., HOUSEKEEPING AND POSITIONING COMMANDS

                     ;RMAS    ATTENTION SUMMARY REGISTER

     000377          ATNMSK  =       377             ;MASK FOR ATTENTION BITS

                     ;RMLA    LOOK AHEAD REGISTER

     002000          SC4     =       BIT10           ;SECTOR COUNT = 16
     001000          SC3     =       BIT09           ;SECTOR COUNT = 8
     000400          SC2     =       BIT08           ;SECTOR COUNT = 4
     000200          SC1     =       BIT07           ;SECTOR COUNT = 2
     000100          SC0     =       BIT06           ;SECTOR COUNT = 1

     003700          SCTMSK  =       003700          ;SECTOR COUNT MASK

                     ;RMMR    MAINTENANCE REGISTER

                     ;        WRITE ONLY BITS

     100000          DBCK    =       BIT15           ;DEBUG CLOCK
     040000          DBEN    =       BIT14           ;DEBUG CLOCK ENABLE
```

```
        020000          DEBL    =       BIT13           ;DIAGNOSTIC END OF BLOCK
        010000          DTO     =       BIT12           ;DIAGNOSTIC TIMEOUT
        004000          MCLK    =       BIT11           ;MAINTENANCE CLOCK
        002000          MRD     =       BIT10           ;READ DATA
        001000          MUR     =       BIT09           ;UNIT READY
        000400          MOC     =       BIT08           ;ON CYLINDER
        000200          MSER    =       BIT07           ;SEEK ERROR
        000100          MDF     =       BIT06           ;DRIVE FAULT
        000040          MS      =       BIT05           ;SECTOR PULSE
        000010          MWP     =       BIT03           ;WRITE PROTECT
        000004          MI      =       BIT02           ;INDEX PULSE
        000002          MSC     =       BIT01           ;SECTOR COMPARE
        000001          DMD     =       BIT00           ;DIAGNOSTIC MODE

                        ;       READ ONLY BITS

        100000          OCC     =       BIT15           ;OCCUPIED
        040000          RG      =       BIT14           ;RUN AND GO
        020000          EBL     =       BIT13           ;END OF BLOCK
        010000          REX     =       BIT12           ;EXCEPTION
        004000          ESRC    =       BIT11           ;ENABLE SEARCH
        002000          PLFS    =       BIT10           ;LOOKING FOR SYNC
        001000          ECRC    =       BIT09           ;ENABLE CRC OUT
        000400          PDA     =       BIT08           ;DATA AREA
        000200          PHA     =       BIT07           ;HEADER AREA
        000100          CONT    =       BIT06           ;CONTINUE
        000040          WC      =       BIT05           ;WORD CLOCK
        000020          EECC    =       BIT04           ;ENABLE ECC OUT
        000010          MWD     =       BIT03           ;WRITE DATA BIT
        000004          LS      =       BIT02           ;LAST SECTOR
        000002          LST     =       BIT01           ;LAST SECTOR AND TRACK
        000001          DMD     =       BIT00           ;DIAGNOSTIC MODE

                        ;RMDT   DRIVE TYPE REGISTER

        100000          NSA     =       BIT15           ;NOT SECTOR ADDRESSED=0
        040000          TAP     =       BIT14           ;TAPE DRIVE = 0
        020000          MOH     =       BIT13           ;MOVING HEAD = 1
        004000          DRQ     =       BIT11           ;DRIVE REQUEST REQUIRED

        020024          SNGPRT  =       020024          ;SINGLE PORT DRIVE TYPE
        024024          DULPRT  =       024024          ;DUAL PORT DRIVE TYPE

                        ;RMOF   OFFSET REGISTER

        010000          FMT16   =       BIT12           ;16 BIT WORD FORMAT
        004000          ECI     =       BIT11           ;ECC INHIBIT
        002000          HCI     =       BIT10           ;HEADER COMPARE INHIBIT
        000200          OFD     =       BIT07           ;OFFSET FORWARD
        161577          XNUOF   =       161577          ;UNUSED BITS OF RMOF


                        ;RMDC   DESIRED CYLINDER ADDRESS REGISTER
        001777          CYLMSK  =       1777            ;MASK FOR CYLINDER ADDRESS
        176000          XNUDC   =       176000          ;UNUSED BITS OF RMDC
```

```
                ;RMMR2  MAINTENANCE REGISTER #2

                ;       READ ONLY BITS
100000          RQA     =       BIT15           ;PORT A REQUEST
040000          RQB     =       BIT14           ;PORT B REQUEST
020000          TAG     =       BIT13           ;TAG CONTROL
010000          TST     =       BIT12           ;COMMAND SEQUENCE TEST BIT
004000          CC      =       BIT11           ;CONTROL OR CYLINDER TAG
002000          CH      =       BIT10           ;CONTROL OR HEAD TAG
001000          BB09    =       BIT09           ;TAG BUS
000400          BB08    =       BIT08           ;TAG BUS
000200          BB07    =       BIT07           ;TAG BUS
000100          BB06    =       BIT06           ;TAG BUS
000040          BB05    =       BIT05           ;TAG BUS
000020          BB04    =       BIT04           ;TAG BUS
000010          BB03    =       BIT03           ;TAG BUS
000004          BB02    =       BIT02           ;TAG BUS
000002          BB01    =       BIT01           ;TAG BUS
000001          BB00    =       BIT00           ;TAG BUS


                ;RMER2  ERROR REGISTER 2

100000          BSE     =       BIT15           ;BAD SECTOR ERROR
040000          SKI     =       BIT14           ;SEEK INCOMPLETE
020000          OPE     =       BIT13           ;OPERATOR PLUG ERROR
010000          IVC     =       BIT12           ;INVALID COMMAND ERROR
004000          LSC     =       BIT11           ;LOSS OF SYSTEM CLOCK
002000          LBC     =       BIT10           ;LOSS OF BIT CLOCK
000200          DVC     =       BIT07           ;DEVICE CHECK
000010          DPE     =       BIT03           ;DATA PARITY ERROR
001567          XNUER2  =       001567          ;UNUSED BITS OF RMER2

                .SBTTL  PROGRAM MNEMONICS

100000          MSE     =       BIT15           ;MANUFACTURING DETECTED SECTOR ERROR
040000          USE     =       BIT14           ;USER DETECTED SECTOR ERROR

                .SBTTL  RM03 REGISTER INDEX VALUES

000000          RMCS1   =       00              ;CONTROL STATUS REGISTER
000006          RMDA    =       06              ;DISK ADDRESS REGISTER
000012          RMDS    =       12              ;DRIVE STATUS REGISTER
000014          RMER1   =       14              ;ERROR REGISTER 1
000016          RMAS    =       16              ;ATTENTION SUMMARY REGISTER
000020          RMLA    =       20              ;LOOK AHEAD REGISTER
000024          RMMR1   =       24              ;MAINTENANCE REGISTER
000026          RMDT    =       26              ;DRIVE TYPE REGISTER
000030          RMSN    =       30              ;SERIAL NUMBER REGISTER
000032          RMOF    =       32              ;OFFSET REGISTER
000034          RMDC    =       34              ;DESIRED CYLINDER REGISTER
000036          RMHR    =       36              ;HOLDING REGISTER
000040          RMMR2   =       40              ;MAINTENANCE REGISTER 2
000042          RMER2   =       42              ;ERROR REGISTER 2
000044          RMEC1   =       44              ;ECC POSITION REGISTER
000046          RMEC2   =       46              ;ECC PATTERN REGISTER
```

```
        000050          ILRG50  =       50              ;ILLEGAL REGISTER 50
        000052          ILRG52  =       52              ;ILLEGAL REGISTER 52
        000054          ILRG54  =       54              ;ILLEGAL REGISTER 54
        000056          ILRG56  =       56              ;ILLEGAL REGISTER 56
        000060          ILRG60  =       60              ;ILLEGAL REGISTER 60
        000062          ILRG62  =       62              ;ILLEGAL REGISTER 62
        000064          ILRG64  =       64              ;ILLEGAL REGISTER 64
        000066          ILRG66  =       66              ;ILLEGAL REGISTER 66
        000070          ILRG70  =       70              ;ILLEGAL REGISTER 70
        000072          ILRG72  =       72              ;ILLEGAL REGISTER 72
        000074          ILRG74  =       74              ;ILLEGAL REGISTER 74
        000076          ILRG76  =       76              ;ILLEGAL REGISTER 76


        000077          IDXMSK  =       77              ;MASK FOR REGISTER INDEX NUMBER

                        .SBTTL  RH CONTROLLER REGISTER BIT DEFINITIONS

                        ;RMCS1  CONTROL STATUS REGISTER #1

        100000          SC      =       BIT15           ;SPECIAL CONDITION-READ ONLY
        040000          TRE     =       BIT14           ;TRANSFER ERROR
        020000          MCPE    =       BIT13           ;MASSBUS CONTROL BUS PARITY
                                                        ;ERROR-READ ONLY
        002000          PSEL    =       BIT10           ;PORT B SELECT
        001000          A17     =       BIT09           ;ADDRESS EXTENSION
        000400          A16     =       BIT08           ;ADDRESS EXTENSION
        000200          RDY     =       BIT07           ;READY-READ ONLY
        000100          IE      =       BIT06           ;INTERRUPT ENABLE

                        ;RMCS2  RH CONTROL STATUS REGISTER #2

        100000          DLT     =       BIT15           ;DATA LATE-READ ONLY
        040000          WCE     =       BIT14           ;WRITE CHECK ERROR-READ ONLY
        020000          UPE     =       BIT13           ;UNIBUS PARITY ERROR
        010000          NED     =       BIT12           ;NONEXISTANT DRIVE-READ ONLY
        004000          NEM     =       BIT11           ;NONEXISTANT MEMORY-READ ONLY
        002000          PGE     =       BIT10           ;PROGRAM ERROR-READ ONLY
        001000          MXF     =       BIT09           ;MISSED TRANSFER
        000400          MDPE    =       BIT08           ;MASSBUS DATA BUS PARITY
                                                        ;ERROR-READ ONLY
        000200          OR      =       BIT07           ;OUTPUT READY-READ ONLY
        000100          IR      =       BIT06           ;INPUT READY-READ ONLY
        000040          CLR     =       BIT05           ;CONTROLLER CLEAR
        000020          PAT     =       BIT04           ;PARITY TEST
        000010          BAI     =       BIT03           ;UNIBUS ADDRESS INCREMENT
                                                        ;INHIBIT
        000004          U2      =       BIT02           ;UNIT SELECT
        000002          U1      =       BIT01           ;UNIT SELECT
        000001          U0      =       BIT00           ;UNIT SELECT

                        ;UNIT SELECT MASK
        000007          UNTMSK  =       7               ;UNIT SELECT MASK

                        ;RMCS3  RH70 CONTROL STATUS REGISTER #3
        100000          APE     =       BIT15           ;ADDRESS PARITY ERROR
```

```
         040000              DPEHI   =       BIT14           ;DATA PARITY ERROR HIGH WORD
         020000              DPELO   =       BIT13           ;DATA PARITY ERROR LOW WORD
         010000              WCEHI   =       BIT12           ;WRITE CHECK ERROR HIGH WORD
         004000              WCELO   =       BIT11           ;WRITE CHECK ERROR LOW WORD
         002000              DBL     =       BIT10           ;DOUBLE WORD TRANSFER
         000100              IE      =       BIT06           ;INTERRUPT ENABLE
         000010              IPCK3   =       BIT03           ;INVERT PARITY CHECK
         000004              IPCK2   =       BIT02           ;INVERT PARITY CHECK
         000002              IPCK1   =       BIT01           ;INVERT PARITY CHECK
         000001              IPCK0   =       BIT00           ;INVERT PARITY CHECK
                             .SBTTL  RH CONTROLLER REGISTER INDEX VALUES

         000000              RMCS1   =       00              ;CONTROL, STATUS REGISTER
         000002              RMWC    =       02              ;WORD COUNT REGISTER
         000004              RMBA    =       04              ;BUS ADDRESS REGISTER
         000010              RMCS2   =       10              ;CONTROLLER STATUS REGISTER
         000022              RMDB    =       22              ;DATA BUFFER
         000050              RMBAE   =       50              ;BUS ADDRESS EXTENSION
         000052              RMCS3   =       52              ;CONTROL STATUS REGISTER #3

         176700              ABASE   =       176700          ;UNIBUS ADDRESS
         120254              AVECT1  =       120254          ;UNIBUS VECTOR ADDRESS AND PRIORITY


                             .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  ACT11 HOOKS

                             ;;***************************************************************
                             ;HOOKS REQUIRED BY ACT11
         000200                      $SVPC=.                 ;SAVE PC
         000046                      .=46
000046   057274                      $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
         000052                      .=52
000052   000000                      .WORD   0               ;;2)SET LOC.52 TO ZERO
         000200                      .=$SVPC                 ;; RESTORE PC

                             .SBTTL  STARTING ADDRESS

                             ;THE PROGRAM STARTS AT LOCATION 200
         000200              .       =       200
000200   000137  004542              JMP     START           ;JUMP TO START OF PROGRAM

         001100              .=1100
                             .SBTTL  APT PARAMETER BLOCK

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

```
3155                            ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
3156                            ;;*************************************************************
3157                  001100            .$X=.      ;;SAVE CURRENT LOCATION
3158                  000024            .=24       ;;SET POWER FAIL TO POINT TO START OF PROGRAM
3159    000024        000200            200        ;;FOR APT START UP
3160                  000044            .=44       ;;POINT TO APT INDIRECT ADDRESS PNTR.
3161    000044        001100            $APTHDR    ;;POINT TO APT HEADER BLOCK
3162                  001100            .=.$X      ;;RESET LOCATION COUNTER
3163                            ;;*************************************************************
3164                            ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
3165                            ;INTERFACE SPEC.
3166
3167    001100                  $APTHD:
3168    001100        000000    $HIBTS: .WORD   0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
3169    001102        001222    $MBADR: .WORD   $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
3170    001104        000001    $TSTM:  .WORD   1          ;;RUN TIM OF LONGEST TEST
3171    001106        000002    $PASTM: .WORD   2          ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
3172    001110        000002    $UNITM: .WORD   2          ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
3173    001112        000042            .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
3174                  001114    TAGADR  =       .
```

```
                                .SBTTL  COMMON TAGS

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

                001114                  .=TAGADR
        001114                  $CMTAG:                                 ;;START OF COMMON TAGS
        001114  000000                  .WORD   0
        001116     000          $TSTNM: .BYTE   0                       ;;CONTAINS THE TEST NUMBER
        001117     000          $ERFLG: .BYTE   0                       ;;CONTAINS ERROR FLAG
        001120  000000          $ICNT:  .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
        001122  000000          $LPADR: .WORD   0                       ;;CONTAINS SCOPE LOOP ADDRESS
        001124  000000          $LPERR: .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
        001126  000000          $ERTTL: .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
        001130     000          $ITEMB: .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
        001131     001          $ERMAX: .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
        001132  000000          $ERRPC: .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
        001134  000000          $GDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'GOOD' DATA
        001136  000000          $BDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'BAD' DATA
        001140  000000          $GDDAT: .WORD   0                       ;;CONTAINS 'GOOD' DATA
        001142  000000          $BDDAT: .WORD   0                       ;;CONTAINS 'BAD' DATA
        001144  000000                  .WORD   0                       ;;RESERVED--NOT TO BE USED
        001146  000000                  .WORD   0
        001150     000          $AUTOB: .BYTE   0                       ;;AUTOMATIC MODE INDICATOR
        001151     000          $INTAG: .BYTE   0                       ;;INTERRUPT MODE INDICATOR
        001152  000000                  .WORD   0
        001154  177570          SWR:    .WORD   DSWR                    ;;ADDRESS OF SWITCH REGISTER
        001156  177570          DISPLAY: .WORD  DDISP                   ;;ADDRESS OF DISPLAY REGISTER
        001160  177560          $TKS:   177560                          ;;TTY KBD STATUS
        001162  177562          $TKB:   177562                          ;;TTY KBD BUFFER
        001164  177564          $TPS:   177564                          ;;TTY PRINTER STATUS REG. ADDRESS
        001166  177566          $TPB:   177566                          ;;TTY PRINTER BUFFER REG. ADDRESS
        001170     000          $NULL:  .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
        001171     002          $FILLS: .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
        001172     012          $FILLC: .BYTE   12                      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
        001173     000          $TPFLG: .BYTE   0                       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
        001174  000000          $TMP0:  .WORD   0                       ;;USER DEFINED
        001176  000000          $TMP1:  .WORD   0                       ;;USER DEFINED
        001200  000000          $TMP2:  .WORD   0                       ;;USER DEFINED
        001202  000000          $TMP3:  .WORD   0                       ;;USER DEFINED
        001204  000000          $TMP4:  .WORD   0                       ;;USER DEFINED
        001206  000000          $TIMES: 0                               ;;MAX. NUMBER OF ITERATIONS
        001210  000000          $ESCAPE:0                               ;;ESCAPE ON ERROR ADDRESS
        001212  177607  000377  $BELL:  .ASCIZ  <207><377><377>         ;;CODE FOR BELL
        001216     077          $QUES:  .ASCII  /?/                     ;;QUESTION MARK
        001217     015          $CRLF:  .ASCII  <15>                    ;;CARRIAGE RETURN
        001220  000012          $LF:    .ASCIZ  <12>                    ;;LINE FEED
                                ;;***********************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;***********************************************************
                                .EVEN
        001222                  $MAIL:                  ;;APT MAILBOX
        001222  000000          $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
        001224  000000          $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
```

```
001226  000000      $TESTN: .WORD   ATESTN  ;;TEST NUMBER
001230  000000      $PASS:  .WORD   APASS   ;;PASS COUNT
001232  000000      $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
001234  000000      $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
001236  000000      $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
001240  000000      $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
001242              $ETABLE:                ;;APT ENVIRONMENT TABLE
001242     000      $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
001243     000      $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
001244  000000      $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
001246  000000      $USWR:  .WORD   AUSWR   ;;USER SWITCHES
001250  000000      $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                    ;*                      BITS 15-11=CPU TYPE
                    ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                    ;*                              11/70=06,PDQ=07,Q=10
                    ;*                      BIT 10=REAL TIME CLOCK
                    ;*                      BIT  9=FLOATING POINT PROCESSOR
                    ;*                      BIT  8=MEMORY MANAGEMENT
001252     000      $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
001253     000      $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                    ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                    ;*                              900 NSEC CORE=001
                    ;*                              300 NSEC BIPOLAR=002
                    ;*                              500 NSEC MOS=003
001254  000000      $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                    ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001256     000      $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001257     000      $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001260  000000      $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001262     000      $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001263     000      $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001264  000000      $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001266     000      $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001267     000      $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001270  000000      $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001272  120254      $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001274  000000      $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001276  176700      $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001300  000000      $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001302  000000      $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
001304  000000      $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
001306  000000      $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
001310  000000      $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
001312  000000      $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
001314  000000      $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001316  000000      $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001320  000000      $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
001322  000000      $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001324  000000      $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001326              $ETEND:
                    .MEXIT



                    .SBTTL  REGISTER INPUT BUFFER
```

```
                            ;       THE REGISTER INPUT BUFFER IS USED FOR STORING REGISTER
                            ;CONTENTS AS THEY ARE READ FROM THE DEVICE.

 001326  000000             RMCS1I: .WORD                   ;CONTROL,STATUS REGISTER #1
 001330  000000             RMWCI: .WORD                    ;WORD COUNT REGISTER
 001332  000000             RMBAI: .WORD                    ;BUS ADDRESS REGISTER
 001334  000000             RMDAI: .WORD                    ;DISK ADDRESS REGISTER
 001336  000000             RMCS2I: .WORD                   ;CONTROL, STATUS REGISTER #2
 001340  000000             RMDSI: .WORD                    ;DRIVE STATUS REGISTER
 001342  000000             RMER1I: .WORD                   ;ERROR REGISTER 1
 001344  000000             RMASI: .WORD                    ;ATTENTION SUMMARY REGISTER
 001346  000000             RMLAI: .WORD                    ;LOOK AHEAD REGISTER
 001350  000000             RMDBI: .WORD                    ;DATA BUFFER
 001352  000000             RMMR1I: .WORD                   ;MAINTENANCE REGISTER #1
 001354  000000             RMDTI: .WORD                    ;DRIVE TYPE REGISTER
 001356  000000             RMSNI: .WORD                    ;SERIAL NUMBER REGISTER
 001360  000000             RMOFI: .WORD                    ;OFFSET REGISTER
 001362  000000             RMDCI: .WORD                    ;DESIRED CYLINDER REGISTER
 001364  000000             RMHRI: .WORD                    ;HOLDING REGISTER
 001366  000000             RMMR2I: .WORD                   ;MAINTENANCE REGISTER #2
 001370  000000             RMER2I: .WORD                   ;ERROR REGISTER 2
 001372  000000             RMEC1I: .WORD                   ;ECC POSITION REGISTER
 001374  000000             RMEC2I: .WORD                   ;ECC PATTERN REGISTER
 001376  000000             RMBAEI: .WORD           ;BUS ADDRESS EXTENSION REGISTER
 001400  000000             RMCS3I: .WORD           ;CONTROL, STATUS REGISTER #3



                            .SBTTL  REGISTER OUTPUT BUFFER

                            ;       THE REGISTER OUTPUT BUFFER IS USED FOR ASSEMBLING DATA TO
                            ;BE WRITTEN TO THE DEVICE.

 001402  000000             RMCS1O: .WORD                   ;CONTROL,STATUS REGISTER #1
 001404  000000             RMWCO: .WORD                    ;WORD COUNT REGISTER
 001406  000000             RMBAO: .WORD                    ;BUS ADDRESS REGISTER
 001410  000000             RMDAO: .WORD                    ;DISK ADDRESS REGISTER
 001412  000000             RMCS2O: .WORD                   ;CONTROL, STATUS REGISTER #2
 001414  000000             RMDSO: .WORD                    ;DRIVE STATUS REGISTER
 001416  000000             RMER1O: .WORD                   ;ERROR REGISTER 1
 001420  000000             RMASO: .WORD                    ;ATTENTION SUMMARY REGISTER
 001422  000000             RMLAO: .WORD                    ;LOOK AHEAD REGISTER
 001424  000000             RMDBO: .WORD                    ;DATA BUFFER
 001426  000000             RMMR1O: .WORD                   ;MAINTENANCE REGISTER #1
 001430  000000             RMDTO: .WORD                    ;DRIVE TYPE REGISTER
 001432  000000             RMSNO: .WORD                    ;SERIAL NUMBER REGISTER
 001434  000000             RMOFO: .WORD                    ;OFFSET REGISTER
 001436  000000             RMDCO: .WORD                    ;DESIRED CYLINDER REGISTER
 001440  000000             RMHRO: .WORD                    ;HOLDING REGISTER
 001442  000000             RMMR2O: .WORD                   ;MAINTENANCE REGISTER #2
 001444  000000             RMER2O: .WORD                   ;ERROR REGISTER 2
 001446  000000             RMEC1O: .WORD                   ;ECC POSITION REGISTER
 001450  000000             RMEC2O: .WORD                   ;ECC PATTERN REGISTER
 001452  000000             RMBAEO: .WORD           ;BUS ADDRESS EXTENSION REGISTER
 001454  000000             RMCS3O: .WORD           ;CONTROL, STATUS REGISTER #3
```

```
                        .SBTTL  TEST QUE

                        ;       EACH WORD OF THE TEST QUE CONTAINS THE DEVICE NUMBER IN
                        ;THE LOW BYTE AND THE ATTENTION BIT IN THE HIGH BYTE. A ZERO
                        ;WORD IS A BLANK AND REPRESENTS THE END OF THE QUE.

001456  000012          TSTQUE: .BLKW   10.     ;TEST QUE

001502  172540          $LPCSR: .WORD   172540          ;KW11-P CONTROL + STATUS REGISTER
001504  172542          $LPCSB: .WORD   172542          ;KW11-P COUNT SET BUFFER
001506  000104          $LPVEC: .WORD   104             ;KW11-P INTERRUPT VECTOR
001510  000106                  .WORD   106
001512  177546          $LLCSR: .WORD   177546          ;KW11-L CONTROL + STATUS REGISTER
001514  000100          $LLVEC: .WORD   100             ;KW11-L INTERRUPT VECTOR
001516  000102                  .WORD   102
001520  000000          $PSW:   .WORD                   ;STORAGE FOR PRIORITY
001522  000000          TIME:   .WORD                   ;STORAGE FOR ELAPSED TIME
001524  000000          WATCH:  .WORD                   ;STORAGE FOR REMAINING TIME
001526  000000          CLOCK:  .WORD                   ;ADDRESS OF START CLOCK SUB
001530  000000          STOP:   .WORD                   ;ADDRESS OF STOP CLOCK SUB

                        ;PUT TAGS HERE
```

```
                        .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


 001532                 $ERRTB:
```

```

                                 ;ERROR  1       CANNOT CLEAR NED STATUS

        001532  066522                   EMT1
        001534  074400                   EHT1
        001536  074474                   EDT1
        001540  074522                   EFT1


                                 ;ERROR  2       CANNOT READ OR WRITE ANY DEVICE REG WITHOUT NED

        001542  066530                   EMT2
        001544  074404                   EHT2
        001546  074476                   EDT2
        001550  074524                   EFT2


                                 ;ERROR  3       CANNOT WRITE/READ ONES TO ANY DEVICE REGISTER

        001552  066556                   EMT3
        001554  000000                   0
        001556  000000                   0
        001560  000000                   0


                                 ;ERROR  4       CANNOT CLEAR ANY DEVICE REGISTER BITS W/MASSBUS INIT

        001562  066576                   EMT4
        001564  000000                   0
        001566  000000                   0
        001570  000000                   0


                                 ;ERROR  5       CANNOT WRITE/READ ZEROS TO ALL BIT POSITIONS

        001572  066620                   EMT5
        001574  074410                   EHT5
        001576  074500                   EDT5
        001600  074526                   EFT5


                                 ;ERROR  6       CANNOT WRITE/READ ONES TO ALL BIT POSITIONS

        001602  066644                   EMT6
        001604  074410                   EHT5
        001606  074500                   EDT5
        001610  074526                   EFT5


                                 ;ERROR  7       CANNOT WRITE/READ SHIFTING ONE BIT TO ALL BIT POSITIONS
                                 ;               OF DEVICE REGISTERS

        001612  066666                   EMT7
        001614  074414                   EHT7
        001616  074500                   EDT5
        001620  074526                   EFT5
```

```


                                          ;ERROR  10      REGISTER SELECT 1 APPEARS S-A-0

    001622  066710                                EMT10
    001624  000000                                0
    001626  000000                                0
    001630  000000                                0


                                          ;ERROR  11      REGISTER SELECT 1 APPEARS S-A-1

    001632  066726                                EMT11
    001634  000000                                0
    001636  000000                                0
    001640  000000                                0


                                          ;ERROR  12      REGISTER SELECT 2 APPEARS S-A-0

    001642  066744                                EMT12
    001644  000000                                0
    001646  000000                                0
    001650  000000                                0


                                          ;ERROR  13      REGISTER SELECT 2 APPEARS S-A-1

    001652  066762                                EMT13
    001654  000000                                0
    001656  000000                                0
    001660  000000                                0


                                          ;ERROR  14      REGISTER SELECT 4 APPEARS S-A-0

    001662  067000                                EMT14
    001664  000000                                0
    001666  000000                                0
    001670  000000                                0


                                          ;ERROR  15      REGISTER SELECT 4 APPEARS S-A-1

    001672  067016                                EMT15
    001674  000000                                0
    001676  000000                                0
    001700  000000                                0


                                          ;ERROR  16      REGISTER SELECT 8 APPEARS S-A-0

    001702  067034                                EMT16
    001704  000000                                0
    001706  000000                                0
    001710  000000                                0
```

```


                               ;ERROR  17       REGISTER SELECT 8 APPEARS S-A-1

        001712  067052                  EMT17
        001714  000000                  0
        001716  000000                  0
        001720  000000                  0


                               ;ERROR  20       CANT WRITE ZEROS RMDA

        001722  067070                  EMT20
        001724  074400                  EHT1
        001726  074474                  EDT1
        001730  074522                  EFT1


                               ;ERROR  21       CANT WRITE ONES RMDA

        001732  067110                  EMT21
        001734  074400                  EHT1
        001736  074474                  EDT1
        001740  074522                  EFT1


                               ;ERROR  22       BIT INTERFERENCE IN WRITING/READING RMDA

        001742  067130                  EMT22
        001744  074400                  EHT1
        001746  074474                  EDT1
        001750  074522                  EFT1


                               ;ERROR  23       CANT WRITE ZEROS RMCS1

        001752  067144                  EMT23
        001754  074400                  EHT1
        001756  074474                  EDT1
        001760  074522                  EFT1


                               ;ERROR  24       CANT WRITE ONES RMCS1

        001762  067164                  EMT24
        001764  074400                  EHT1
        001766  074474                  EDT1
        001770  074522                  EFT1


                               ;ERROR  25       BIT INTERFERENCE IN WRITING/READING RMCS1

        001772  067204                  EMT25
        001774  074400                  EHT1
        001776  074474                  EDT1
        002000  074522                  EFT1
```

```


                                 ;ERROR  26       MBA CLR L IS STUCK ACTIVE

       002002  067220                    EMT26
       002004  000000                    0
       002006  000000                    0
       002010  000000                    0


                                 ;ERROR  27       CANNOT CLEAR RMER1-PAR,RMR,ILF,ILR

       002012  067252                    EMT27
       002014  074400                    EHT1
       002016  074474                    EDT1
       002020  074522                    EFT1


                                 ;ERROR  30       CANNOT CLEAR RMER1-DCK,IAE,AOE,HCRC,HCE,ECH,WCF,FER

       002022  067264                    EMT30
       002024  074400                    EHT1
       002026  074474                    EDT1
       002030  074522                    EFT1


                                 ;ERROR  31       CANNOT CLEAR RMER1-OPI,DTE

       002032  067300                    EMT31
       002034  074400                    EHT1
       002036  074474                    EDT1
       002040  074522                    EFT1


                                 ;ERROR  32       CANNOT WRITE 0 IN RMER1-PAR,RMR,ILF,ILR

       002042  067314                    EMT32
       002044  074400                    EHT1
       002046  074474                    EDT1
       002050  074522                    EFT1


                                 ;ERROR  33       CANNOT WRITE 0 IN RMER1-DCK,IAE,AOE,HCRC,HCE,ECH,WCF,FER

       002052  067330                    EMT33
       002054  074400                    EHT1
       002056  074474                    EDT1
       002060  074522                    EFT1


                                 ;ERROR  34       CANNOT WRITE 0 IN RMER1-OPI,DTE

       002062  067346                    EMT34
       002064  074400                    EHT1
       002066  074474                    EDT1
       002070  074522                    EFT1
```

```


                                ;ERROR  35      CANNOT WRITE 1 IN RMER1

002072  067364                          EMT35
002074  074400                          EHT1
002076  074474                          EDT1
002100  074522                          EFT1


                                ;ERROR  36      CANNOT WRITE SHIFTING 1 IN RMER1

002102  067400                          EMT36
002104  074400                          EHT1
002106  074474                          EDT1
002110  074522                          EFT1


                                ;ERROR  37      CANNOT WRITE ZEROS IN RMDC

002112  067414                          EMT37
002114  074400                          EHT1
002116  074474                          EDT1
002120  074522                          EFT1


                                ;ERROR  40      CANNOT WRITE ONES IN RMDC

002122  067430                          EMT40
002124  074400                          EHT1
002126  074474                          EDT1
002130  074522                          EFT1


                                ;ERROR  41      BIT INTERFERENCE IN WRITING/READING RMDC

002132  067450                          EMT41
002134  074400                          EHT1
002136  074474                          EDT1
002140  074522                          EFT1


                                ;ERROR  42      CANNOT WRITE 1'S IN RMDC OR RMDA

002142  067464                          EMT42
002144  000000                          0
002146  000000                          0
002150  000000                          0


                                ;ERROR  43      CANNOT CLEAR RMCS1-FUNCTION CODE

002152  067510                          EMT43
002154  074400                          EHT1
002156  074474                          EDT1
002160  074522                          EFT1
```

```


                        ;ERROR  44      UNUSED BITS OF RMER2 NOT ZERO

002162  067522                  EMT44
002164  074400                  EHT1
002166  074474                  EDT1
002170  074522                  EFT1


                        ;ERROR  45      CANNOT CLEAR RMER2-OPE,IVC,LSC

002172  067536                  EMT45
002174  074400                  EHT1
002176  074474                  EDT1
002200  074522                  EFT1


                        ;ERROR  46      CANNOT CLEAR RMER2-LBC,DPE

002202  067552                  EMT46
002204  074400                  EHT1
002206  074474                  EDT1
002210  074522                  EFT1


                        ;ERROR  47      CANNOT WRITE ZEROS RMER2-OPE,IVC,LSC

002212  067566                  EMT47
002214  074400                  EHT1
002216  074474                  EDT1
002220  074522                  EFT1


                        ;ERROR  50      CANNOT WRITE ZEROS RMER2-LBC,DPE

002222  067610                  EMT50
002224  074400                  EHT1
002226  074474                  EDT1
002230  074522                  EFT1


                        ;ERROR  51      CANNOT WRITE ONES RMER2

002232  067632                  EMT51
002234  074400                  EHT1
002236  074474                  EDT1
002240  074522                  EFT1


                        ;ERROR  52      CANNOT WRITE SHIFTING ONES RMER2

002242  067652                  EMT52
002244  074400                  EHT1
002246  074474                  EDT1
002250  074522                  EFT1
```

```
                                        ;ERROR  53        UNUSED BITS OF RMOF ARE NOT ZERO

        002252  067666                          EMT53
        002254  074400                          EHT1
        002256  074474                          EDT1
        002260  074522                          EFT1


                                        ;ERROR  54        CANNOT WRITE ZEROS RMOF-FMT,ECI,HCI,OFD

        002262  067702                          EMT54
        002264  074400                          EHT1
        002266  074474                          EDT1
        002270  074522                          EFT1


                                        ;ERROR  55        CANNOT WRITE ONES RMOF-FMT,ECI,HCI,OFD

        002272  067722                          EMT55
        002274  074400                          EHT1
        002276  074474                          EDT1
        002300  074522                          EFT1


                                        ;ERROR  56        CANNOT WRITE SHIFTING ONES RMOF

        002302  067742                          EMT56
        002304  074400                          EHT1
        002306  074474                          EDT1
        002310  074522                          EFT1


                                        ;ERROR  57        DEVICE IS NOT AN RM03

        002312  067756                          EMT57
        002314  074420                          EHT57
        002316  074502                          EDT57
        002320  074530                          EFT57


                                        ;ERROR  60        DEVICE AVAILABLE IS NOT SET

        002322  067772                          EMT60
        002324  074400                          EHT1
        002326  074474                          EDT1
        002330  074522                          EFT1


                                        ;ERROR  61        CANNOT WRITE ZEROS RMHR

        002332  070006                          EMT61
        002334  074400                          EHT1
        002336  074474                          EDT1
        002340  074522                          EFT1
```

```

                                ;ERROR  62      CANNOT WRITE ONES RMHR

002342  070026                          EMT62
002344  074400                          EHT1
002346  074474                          EDT1
002350  074522                          EFT1


                                ;ERROR  63      CANNOT WRITE SHIFTING ONES RMHR

002352  070046                          EMT63
002354  074400                          EHT1
002356  074474                          EDT1
002360  074522                          EFT1


                                ;ERROR  64      CANNOT CLEAR ILR STATUS

002362  070062                          EMT64
002364  074400                          EHT1
002366  074474                          EDT1
002370  074522                          EFT1


                                ;ERROR  65      ILR ERROR SHOULD NOT BE SET

002372  070074                          EMT65
002374  074424                          EHT65
002376  074504                          EDT65
002400  074532                          EFT65


                                ;ERROR  66      ILR ERROR SHOULD BE SET

002402  070110                          EMT66
002404  074424                          EHT65
002406  074504                          EDT65
002410  074532                          EFT65


                                ;ERROR  67      CANNOT CLEAR PAR STATUS-DPE IS RESET

002412  070124                          EMT67
002414  074400                          EHT1
002416  074474                          EDT1
002420  074522                          EFT1


                                ;ERROR  70      CANNOT CLEAR PAR AND DPE STATUS

002422  070142                          EMT70
002424  074400                          EHT1
002426  074474                          EDT1
002430  074522                          EFT1
```

```

                              ;ERROR  71       "PAR" ERROR SHOULD NOT BE SET-"PAT" IS OFF

        002432  070162                EMT71
        002434  074430                EHT71
        002436  074506                EDT71
        002440  074534                EFT71


                              ;ERROR  72       "PAR" ERROR SHOULD BE SET-"PAT" IS ON

        002442  070206                EMT72
        002444  074430                EHT71
        002446  074506                EDT71
        002450  074534                EFT71


                              ;ERROR  73       "MCPE" ERROR SHOULD NOT BE SET

        002452  070232                EMT73
        002454  074430                EHT71
        002456  074506                EDT71
        002460  074534                EFT71


                              ;ERROR  74       UNEXPECTED BUS TIMEOUT

        002462  070252                EMT74
        002464  074434                EHT74
        002466  074510                EDT74
        002470  074536                EFT74


                              ;ERROR  75       CANT CLEAR "DMD"

        002472  070262                EMT75
        002474  074400                EHT1
        002476  074474                EDT1
        002500  074522                EFT1


                              ;ERROR  76       CANT WRITE ZERO "DMD"

        002502  070274                EMT76
        002504  074400                EHT1
        002506  074474                EDT1
        002510  074522                EFT1


                              ;ERROR  77       CANT WRITE ONE "DMD"

        002512  070310                EMT77
        002514  074400                EHT1
        002516  074474                EDT1
        002520  074522                EFT1
```

```
                                ;ERROR  100     DMD SET BY WRONG BIT

002522  070324                          EMT100
002524  074430                          EHT71
002526  074504                          EDT65
002530  074532                          EFT65

                                ;ERROR  101     CANT CLEAR "MOL" IN DIAGNOSTIC MODE

002532  070344                          EMT101
002534  074400                          EHT1
002536  074474                          EDT1
002540  074522                          EFT1

                                ;ERROR  102     CANT SET "MOL" IN DIAGNOSTIC MODE

002542  070364                          EMT102
002544  074400                          EHT1
002546  074474                          EDT1
002550  074522                          EFT1

                                ;ERROR  103     "MUR" SET BY WRONG BIT

002552  070404                          EMT103
002554  074430                          EHT71
002556  074504                          EDT65
002560  074532                          EFT65

                                ;ERROR  104     CANT RESET "WRL" IN DIAGNOSTIC MODE

002562  070430                          EMT104
002564  074400                          EHT1
002566  074474                          EDT1
002570  074522                          EFT1

                                ;ERROR  105     CANT SET "WRL" IN DIAGNOSTIC MODE

002572  070450                          EMT105
002574  074400                          EHT1
002576  074474                          EDT1
002600  074522                          EFT1

                                ;ERROR  106     "MWP" SET BY WRONG BIT

002602  070470                          EMT106
002604  074430                          EHT71
002606  074504                          EDT65
002610  074532                          EFT65
```

```

                                        ;ERROR  107     CANT RESET "DVC" USING "MDVC"

   002612  070514                               EMT107
   002614  074400                               EHT1
   002616  074474                               EDT1
   002620  074522                               EFT1


                                        ;ERROR  110     "DVC" IS RESET BUT "UNS" IS SET

   002622  070534                               EMT110
   002624  074400                               EHT1
   002626  074474                               EDT1
   002630  074522                               EFT1


                                        ;ERROR  111     "DVC" IS SET BUT "UNS" IS NOT SET

   002632  070556                               EMT111
   002634  074400                               EHT1
   002636  074474                               EDT1
   002640  074522                               EFT1


                                        ;ERROR  112     CANT SET "DVC" USING MDVC"

   002642  070576                               EMT112
   002644  074400                               EHT1
   002646  074474                               EDT1
   002650  074522                               EFT1


                                        ;ERROR  113     "DVC" IS RESET BUT "UNS" IS SET

   002652  070616                               EMT113
   002654  074400                               EHT1
   002656  074474                               EDT1
   002660  074522                               EFT1


                                        ;ERROR  114     "DVC" IS SET BUT "UNS" IS NOT SET

   002662  070642                               EMT114
   002664  074400                               EHT1
   002666  074474                               EDT1
   002670  074522                               EFT1


                                        ;ERROR  115     "MDF" IS SET BY WRONG BIT

   002672  070664                               EMT115
   002674  074440                               EHT115
   002676  074512                               EDT115
   002700  074540                               EFT115
```

```


                                ;ERROR  116     CANT RESET "SKI" USING "MSER"

002702  070710                          EMT116
002704  074400                          EHT1
002706  074474                          EDT1
002710  074522                          EFT1


                                ;ERROR  117     CANT SET "SKI" USING "MSER"

002712  070730                          EMT117
002714  074400                          EHT1
002716  074474                          EDT1
002720  074522                          EFT1


                                ;ERROR  120     "SKI" SET BY WRONG BIT

002722  070750                          EMT120
002724  074440                          EHT115
002726  074512                          EDT115
002730  074540                          EFT115


                                ;ERROR  121     CANT RESET "PIP" USING "MOC"

002732  070774                          EMT121
002734  074400                          EHT1
002736  074474                          EDT1
002740  074522                          EFT1


                                ;ERROR  122     CANT SET "PIP" USING "MOC"

002742  071014                          EMT122
002744  074400                          EHT1
002746  074474                          EDT1
002750  074522                          EFT1


                                ;ERROR  123     "MOC" SET BY WRONG BIT

002752  071034                          EMT123
002754  074440                          EHT115
002756  074512                          EDT115
002760  074540                          EFT115


                                ;ERROR  124     CANT CLEAR "EBL"

002762  071060                          EMT124
002764  074400                          EHT1
002766  074474                          EDT1
002770  074522                          EFT1
```

```


                              ;ERROR  125     "EBL" NOT ZERO IN DIAGNOSTIC MODE

002772  071076                        EMT125
002774  074400                        EHT1
002776  074474                        EDT1
003000  074522                        EFT1


                              ;ERROR  126     CANT SET "EBL" USING "DEBL"

003002  071116                        EMT126
003004  074400                        EHT1
003006  074474                        EDT1
003010  074522                        EFT1


                              ;ERROR  127     "DEBL" SET BY WRONG BIT

003012  071134                        EMT127
003014  074440                        EHT115
003016  074512                        EDT115
003020  074540                        EFT115


                              ;ERROR  130     "LS" NOT CORRECT ACCORDING TO RMDA

003022  071160                        EMT130
003024  074444                        EHT130
003026  074514                        EDT130
003030  074542                        EFT130


                              ;ERROR  131     "LST" NOT CORRECT ACCORDING TO RMDA

003032  071176                        EMT131
003034  074444                        EHT130
003036  074514                        EDT130
003040  074542                        EFT130


                              ;ERROR  132     CANNOT INCREMENT SECTOR ADDRESS USING "DEBL"

003042  071214                        EMT132
003044  074450                        EHT132
003046  074516                        EDT132
003050  074544                        EFT132


                              ;ERROR  133     CANNOT INCREMENT TRACK ADDRESS USING "DEBL"

003052  071234                        EMT133
003054  074450                        EHT132
003056  074516                        EDT132
003060  074544                        EFT132
```

```
                                ;ERROR  134     UNUSED BITS OF RMDC NOT ZERO

        003062  071254                  EMT134
        003064  074400                  EHT1
        003066  074474                  EDT1
        003070  074522                  EFT1


                                ;ERROR  135     "VV" NOT RESET BY UNIT READY

        003072  071270                  EMT135
        003074  074400                  EHT1
        003076  074474                  EDT1
        003100  074522                  EFT1


                                ;ERROR  136     SERIAL NUMBER IS INCONSISTENT

        003102  071306                  EMT136
        003104  074400                  EHT1
        003106  074474                  EDT1
        003110  074522                  EFT1


                                ;ERROR  137     CANT CLEAR "GO" BIT

        003112  071320                  EMT137
        003114  074400                  EHT1
        003116  074474                  EDT1
        003120  074522                  EFT1


                                ;ERROR  140     CANT INCREMENT CYLINDER USING "DEBL"

        003122  071336                  EMT140
        003124  074450                  EHT132
        003126  074516                  EDT132
        003130  074544                  EFT132


                                ;ERROR  141     CANT RESET "LBT" BY WRITING RMDA

        003132  071356                  EMT141
        003134  074400                  EHT1
        003136  074474                  EDT1
        003140  074522                  EFT1


                                ;ERROR  142     CANT SET "LBT" USING "DEBL"

        003142  071372                  EMT142
        003144  074400                  EHT1
        003146  074474                  EDT1
        003150  074522                  EFT1
```

```
                                ;ERROR  143     CANT READ ZERO FROM COMP ERROR

        003152  071410                  EMT143
        003154  074400                  EHT1
        003156  074474                  EDT1
        003160  074522                  EFT1

                                ;ERROR  144     CANT SET COMP ERROR WITH RMER1 OR RMER2

        003162  071424                  EMT144
        003164  074400                  EHT1
        003166  074474                  EDT1
        003170  074522                  EFT1

                                ;ERROR  145     COMP ERROR DID NOT SET

        003172  071446                  EMT145
        003174  074454                  EHT145
        003176  074514                  EDT130
        003200  074542                  EFT130

                                ;ERROR  146     CANT SET "GO" BIT

        003202  071470                  EMT146
        003204  074400                  EHT1
        003206  074474                  EDT1
        003210  074522                  EFT1

                                ;ERROR  147     CANT READ A ONE FROM "TST"

        003212  071506                  EMT147
        003214  074400                  EHT1
        003216  074474                  EDT1
        003220  074522                  EFT1

                                ;ERROR  150     "TST" IS INCORRECT FOR THE FUNCTION CODE

        003222  071520                  EMT150
        003224  074460                  EHT150
        003226  074512                  EDT115
        003230  074540                  EFT115

                                ;ERROR  151     CANT SET THE "GO" BIT

        003232  071542                  EMT151
        003234  074400                  EHT1
        003236  074474                  EDT1
        003240  074522                  EFT1
```

```


                                ;ERROR  152     "DRY" NOT THE COMPLEMENT OF "GO"

        003242  071554                  EMT152
        003244  074400                  EHT1
        003246  074474                  EDT1
        003250  074522                  EFT1


                                ;ERROR  153     "GO" RESET EARLY

        003252  071570                  EMT153
        003254  074400                  EHT1
        003256  074474                  EDT1
        003260  074522                  EFT1


                                ;ERROR  154     "GO" DIDNT RESET ON TIME

        003262  071610                  EMT154
        003264  074400                  EHT1
        003266  074474                  EDT1
        003270  074522                  EFT1


                                ;ERROR  155     CANT CLEAR CONTINUE

        003272  071630                  EMT155
        003274  074400                  EHT1
        003276  074474                  EDT1
        003300  074522                  EFT1


                                ;ERROR  156     CONTINUE IS INCORRECT FOR THE FUNCTION CODE

        003302  071646                  EMT156
        003304  074460                  EHT150
        003306  074512                  EDT115
        003310  074540                  EFT115


                                ;ERROR  157     CANT CLEAR IVC

        003312  071670                  EMT157
        003314  074400                  EHT1
        003316  074474                  EDT1
        003320  074522                  EFT1


                                ;ERROR  160     IVC IS INCORRECT FOR THE FUNCTION CODE

        003322  071706                  EMT160
        003324  074460                  EHT150
        003326  074512                  EDT115
        003330  074540                  EFT115
```

```


                                ;ERROR  161     CANT CLEAR LSC

 003332  071736                         EMT161
 003334  074400                         EHT1
 003336  074474                         EDT1
 003340  074522                         EFT1


                                ;ERROR  162     CANT SET LSC

 003342  071754                         EMT162
 003344  074400                         EHT1
 003346  074474                         EDT1
 003350  074522                         EFT1


                                ;ERROR  163     COMMAND DECODE WAS ENABLED WITH COMP ERROR SET

 003352  071772                         EMT163
 003354  074400                         EHT1
 003356  074474                         EDT1
 003360  074522                         EFT1


                                ;ERROR  164     COMMAND DECODE WAS ENABLED WITH COMP ERROR SET

 003362  072014                         EMT164
 003364  074400                         EHT1
 003366  074474                         EDT1
 003370  074522                         EFT1


                                ;ERROR  165     DECODE DOES NOT SET

 003372  072036                         EMT165
 003374  000000                         0
 003376  000000                         0
 003400  000000                         0


                                ;ERROR  166     CANT CLEAR OCCUPIED

 003402  072062                         EMT166
 003404  074400                         EHT1
 003406  074474                         EDT1
 003410  074522                         EFT1


                                ;ERROR  167     ILF SET WITHOUT GO BIT

 003412  072102                         EMT167
 003414  074400                         EHT1
 003416  074474                         EDT1
 003420  074522                         EFT1
```

```
                              ;ERROR  170     CANT SET VOLUME VALID

003422  072124                        EMT170
003424  074460                        EHT150
003426  074512                        EDT115
003430  074540                        EFT115


                              ;ERROR  171     ILF IS INCORRECT

003432  072136                        EMT171
003434  074460                        EHT150
003436  074512                        EDT115
003440  074540                        EFT115


                              ;ERROR  172     CANT SET OFFSET DIRECTION BIT

003442  072152                        EMT172
003444  074400                        EHT1
003446  074474                        EDT1
003450  074522                        EFT1


                              ;ERROR  173     OCCUPIED IS INCORRECT FOR FUNCTION CODE

003452  072170                        EMT173
003454  074460                        EHT150
003456  074512                        EDT115
003460  074540                        EFT115


                              ;ERROR  174     READ IN PRESET DIDNT CLEAR RMDA, RMDC OR RMOF

003462  072212                        EMT174
003464  000000                        0
003466  000000                        0
003470  000000                        0


                              ;ERROR  175     READ IN PRESET DIDNT CLEAR RMOF

003472  072240                        EMT175
003474  074400                        EHT1
003476  074474                        EDT1
003500  074522                        EFT1


                              ;ERROR  176     READ IN PRESET DIDNT CLEAR RMDA

003502  072256                        EMT176
003504  074400                        EHT1
003506  074474                        EDT1
003510  074522                        EFT1
```

```


                                        ;ERROR  177     READ IN PRESET DIDNT CLEAR RMDC

 003512  072274                                 EMT177
 003514  074400                                 EHT1
 003516  074474                                 EDT1
 003520  074522                                 EFT1


                                        ;ERROR  200     CANT SET OFFSET MODE BY OFFSET COMMAND

 003522  072312                                 EMT200
 003524  074400                                 EHT1
 003526  074474                                 EDT1
 003530  074522                                 EFT1


                                        ;ERROR  201     CANT RESET OFFSET MODE BY RTC COMMAND

 003532  072330                                 EMT201
 003534  074400                                 EHT1
 003536  074474                                 EDT1
 003540  074522                                 EFT1


                                        ;ERROR  202     CANT RESET OFD BY RTC COMMAND

 003542  072346                                 EMT202
 003544  074400                                 EHT1
 003546  074474                                 EDT1
 003550  074522                                 EFT1


                                        ;ERROR  203     CANT RESET OM BY RMDC

 003552  072364                                 EMT203
 003554  074400                                 EHT1
 003556  074474                                 EDT1
 003560  074522                                 EFT1


                                        ;ERROR  204     CANT RESET OM BY EBL

 003562  072406                                 EMT204
 003564  074400                                 EHT1
 003566  074474                                 EDT1
 003570  074522                                 EFT1


                                        ;ERROR  205     RUN AND GO NOT CORRECT FOR FUNCTION CODE

 003572  072426                                 EMT205
 003574  074460                                 EHT150
 003576  074512                                 EDT115
 003600  074540                                 EFT115
```

```


                              ;ERROR  206     CANT SET IAE ERROR

 003602  072446                       EMT206
 003604  000000                       0
 003606  000000                       0
 003610  000000                       0


                              ;ERROR  207     IAE IS INCORRECT FOR FUNCTION CODE

 003612  072476                       EMT207
 003614  074460                       EHT150
 003616  074512                       EDT115
 003620  074540                       EFT115


                              ;ERROR  210     IAE IS INCORRECT FOR RMDA

 003622  072512                       EMT210
 003624  074440                       EHT115
 003626  074512                       EDT115
 003630  074540                       EFT115


                              ;ERROR  211     IAE IS INCORRECT FOR RMDC

 003632  072530                       EMT211
 003634  074440                       EHT115
 003636  074512                       EDT115
 003640  074540                       EFT115


                              ;ERROR  212     CANT SET AOE

 003642  072546                       EMT212
 003644  074400                       EHT1
 003646  074474                       EDT1
 003650  074522                       EFT1


                              ;ERROR  213     RMR SET WHEN WRITING RMAS OR RMCS

 003652  072560                       EMT213
 003654  074464                       EHT213
 003656  074512                       EDT115
 003660  074540                       EFT115


                              ;ERROR  214     CANT SET RMR

 003662  072610                       EMT214
 003664  074464                       EHT213
 003666  074512                       EDT115
 003670  074540                       EFT115
```

```


                                 ;ERROR  215      DRQ IS 0 AND PGM IS 1

        003672  072622                    EMT215
        003674  074400                    EHT1
        003676  074474                    EDT1
        003700  074522                    EFT1


                                 ;ERROR  216      DVA IS NOT SET

        003702  072642                    EMT216
        003704  074400                    EHT1
        003706  074474                    EDT1
        003710  074522                    EFT1


                                 ;ERROR  217      DPR IS NOT SET

        003712  072656                    EMT217
        003714  074400                    EHT1
        003716  074474                    EDT1
        003720  074522                    EFT1


                                 ;ERROR  220      CANT SET PORT REQUEST BY READING RMCS1

        003722  072672                    EMT220
        003724  074470                    EHT220
        003726  074520                    EDT220
        003730  074546                    EFT220


                                 ;ERROR  221      CANT SET PORT REQUEST BY WRITING RMAS

        003732  072710                    EMT221
        003734  074470                    EHT220
        003736  074520                    EDT220
        003740  074546                    EFT220


                                 ;ERROR  222      CANT SET PORT REQUEST BY WRITING RMDA

        003742  072726                    EMT222
        003744  074470                    EHT220
        003746  074520                    EDT220
        003750  074546                    EFT220


                                 ;ERROR  223      CANT RESET PORT REQUEST BY RELEASE COMMAND

        003752  072744                    EMT223
        003754  074470                    EHT220
        003756  074520                    EDT220
        003760  074546                    EFT220
```

```
                            ;ERROR  224     CANT CLEAR ATA BY RMAS

        003762  072762                  EMT224
        003764  074400                  EHT1
        003766  074474                  EDT1
        003770  074522                  EFT1


                            ;ERROR  225     ATA IS RESET BUT RMAS NOT ZERO

        003772  073002                  EMT225
        003774  074400                  EHT1
        003776  074474                  EDT1
        004000  074522                  EFT1


                            ;ERROR  226     CANT RESET ATA BY GO

        004002  073024                  EMT226
        004004  074400                  EHT1
        004006  074474                  EDT1
        004010  074522                  EFT1


                            ;ERROR  227     ATA NOT SET BY UNIT READY

        004012  073042                  EMT227
        004014  074400                  EHT1
        004016  074474                  EDT1
        004020  074522                  EFT1


                            ;ERROR  230     ATA NOT SET BY UNIT READY

        004022  073060                  EMT230
        004024  074400                  EHT1
        004026  074474                  EDT1
        004030  074522                  EFT1


                            ;ERROR  231     ATA NOT SET BY COMP ERROR

        004032  073076                  EMT231
        004034  074400                  EHT1
        004036  074474                  EDT1
        004040  074522                  EFT1


                            ;ERROR  232     ATA SET/DID NOT SET WHEN REGISTER WRITTEN
                            ;               WHILE COMP ERROR WAS SET

        004042  073112                  EMT232
        004044  074464                  EHT213
        004046  074512                  EDT115
```

```
004050  074540                  EFT115


                        ;ERROR  233     ATA NOT SET BY COMMAND SEQUENCER

004052  073134                  EMT233
004054  074460                  EHT150
004056  074512                  EDT115
004060  074540                  EFT115


                        ;ERROR  234     WLE INCORRECT ACCORDING TO FUNCTION CODE

004062  073156                  EMT234
004064  074460                  EHT150
004066  074512                  EDT115
004070  074540                  EFT115


                        ;ERROR  235     CANT CLEAR EXCEPTION

004072  073204                  EMT235
004074  074400                  EHT1
004076  074474                  EDT1
004100  074522                  EFT1


                        ;ERROR  236     CANT SET EXCEPTION

004102  073224                  EMT236
004104  074440                  EHT115
004106  074512                  EDT115
004110  074540                  EFT115


                        ;ERROR  237     CANT CLEAR OPI

004112  073256                  EMT237
004114  074400                  EHT1
004116  074474                  EDT1
004120  074522                  EFT1


                        ;ERROR  240     CANT SET OPI

004122  073256                  EMT237
004124  074460                  EHT150
004126  074512                  EDT115
004130  074540                  EFT115


                        ;ERROR  241     OPI NOT SET DURING RECALIBRATE

004132  073322                  EMT241
004134  074400                  EHT1
004136  074474                  EDT1
```

```
004140  074522                  EFT1


                ;ERROR  242     RECALIBRATE DID NOT ABORT WHEN DRIVE FAULT SET

004142  073346                  EMT242
004144  074400                  EHT1
004146  074474                  EDT1
004150  074522                  EFT1


                ;ERROR  243     OPI SHOULD HAVE SET BECAUSE ON CYLINDER NEVER
                ;               DROPPED DURING RECALIBRATE

004152  073372                  EMT243
004154  074400                  EHT1
004156  074474                  EDT1
004160  074522                  EFT1


                ;ERROR  244     ATA NOT SET DURING RECALIBRATE

004162  073416                  EMT244
004164  074400                  EHT1
004166  074474                  EDT1
004170  074522                  EFT1


                ;ERROR  245     GO RESET EARLY DURING RECALIBRATE

004172  073434                  EMT245
004174  074400                  EHT1
004176  074474                  EDT1
004200  074522                  EFT1


                ;ERROR  246     GO NOT RESET DURING RECALIBRATE

004202  073452                  EMT246
004204  074400                  EHT1
004206  074474                  EDT1
004210  074522                  EFT1


                ;ERROR  247     INCORRECT TAG BUS DURING RECALIBRATE

004212  073476                  EMT247
004214  074400                  EHT1
004216  074474                  EDT1
004220  074522                  EFT1


                ;ERROR  250     OPI SHOULD HAVE SET DURING SEEK BECAUSE UNIT
                ;               READY DROPPED

004222  073514                  EMT250
```

```
004224  074400                  EHT1
004226  074474                  EDT1
004230  074522                  EFT1


                        ;ERROR  251     SEEK DID NOT ABORT WHEN DRIVE FAULT SET

004232  073540                  EMT251
004234  074400                  EHT1
004236  074474                  EDT1
004240  074522                  EFT1


                        ;ERROR  252     OPI SHOULD HAVE SET BECAUSE ON CYLINDER NEVER
                        ;               DROPPED DURING SEEK

004242  073564                  EMT252
004244  074400                  EHT1
004246  074474                  EDT1
004250  074522                  EFT1


                        ;ERROR  253     ATA NOT SET DURING SEEK

004252  073610                  EMT253
004254  074400                  EHT1
004256  074474                  EDT1
004260  074522                  EFT1


                        ;ERROR  254     GO RESET EARLY DURING SEEK

004262  073626                  EMT254
004264  074400                  EHT1
004266  074474                  EDT1
004270  074522                  EFT1


                        ;ERROR  255     GO DID NOT RESET DURING SEEK

004272  073644                  EMT255
004274  074400                  EHT1
004276  074474                  EDT1
004300  074522                  EFT1


                        ;ERROR  256     INCORRECT TAG BUS DURING SEEK

004302  073670                  EMT256
004304  074400                  EHT1
004306  074474                  EDT1
004310  074522                  EFT1


                        ;ERROR  257     OPI NOT SET DURING SEARCH

```

```
004312  073706                        EMT257
004314  074400                        EHT1
004316  074474                        EDT1
004320  074522                        EFT1


                            ;ERROR  260     SEARCH DID NOT ABORT WHEN DRIVE FAULT SET

004322  073732                        EMT260
004324  074400                        EHT1
004326  074474                        EDT1
004330  074522                        EFT1


                            ;ERROR  261     OPI SHOULD HAVE SET BECAUSE ON CYLINDER NEVER
                            ;               DROPPED DURING SEARCH

004332  073756                        EMT261
004334  074400                        EHT1
004336  074474                        EDT1
004340  074522                        EFT1


                            ;ERROR  262     ATA NOT SET DURING SEARCH

004342  074002                        EMT262
004344  074400                        EHT1
004346  074474                        EDT1
004350  074522                        EFT1


                            ;ERROR  263     GO RESET EARLY DURING SEARCH

004352  074020                        EMT263
004354  074400                        EHT1
004356  074474                        EDT1
004360  074522                        EFT1


                            ;ERROR  264     GO DID NOT RESET DURING SEARCH

004362  074036                        EMT264
004364  074400                        EHT1
004366  074474                        EDT1
004370  074522                        EFT1


                            ;ERROR  265     SEARCH ENABLE DIDNT SET DURING SEARCH

004372  074062                        EMT265
004374  074400                        EHT1
004376  074474                        EDT1
004400  074522                        EFT1


                            ;ERROR  266     INCORRECT TAG BUS DURING SEARCH
```

```

004402  074100              EMT266
004404  074400              EHT1
004406  074474              EDT1
004410  074522              EFT1


                   ;ERROR  267    OPI NOT SET BY SEARCH TIMEOUT

004412  074116              EMT267
004414  074400              EHT1
004416  074474              EDT1
004420  074522              EFT1


                   ;ERROR  270    OPI NOT SET DURING DATA COMMAND

004422  074132              EMT270
004424  074400              EHT1
004426  074474              EDT1
004430  074522              EFT1


                   ;ERROR  271    DATA COMMAND DID NOT ABORT WHEN DRIVE FAULT SET

004432  074156              EMT271
004434  074400              EHT1
004436  074474              EDT1
004440  074522              EFT1


                   ;ERROR  272    EBL RESET EARLY DURING DATA COMMAND

004442  074202              EMT272
004444  074400              EHT1
004446  074474              EDT1
004450  074522              EFT1


                   ;ERROR  273    EBL DIDNT RESET ON TIME DURING DATA COMMAND

004452  074220              EMT273
004454  074400              EHT1
004456  074474              EDT1
004460  074522              EFT1


                   ;ERROR  274    GO NOT RESET DURING DATA COMMAND

004462  074236              EMT274
004464  074400              EHT1
004466  074474              EDT1
004470  074522              EFT1


                   ;ERROR  275    RUN AND GO NOT SET DURING DATA COMMAND
```

```

004472  074254                    EMT275
004474  074400                    EHT1
004476  074474                    EDT1
004500  074522                    EFT1


                         ;ERROR  276      INCORRECT TAG BUS DURING DATA COMMAND

004502  074274                    EMT276
004504  074400                    EHT1
004506  074474                    EDT1
004510  074522                    EFT1


                         ;ERROR  277      OPI NOT SET DURING DATA COMMAND WHEN ON
                         ;                CYLINDER DIDNT DROP

004512  074312                    EMT277
004514  074400                    EHT1
004516  074474                    EDT1
004520  074522                    EFT1


                         ;ERROR  300      DATA COMMAND DID NOT ABORT WHEN SEEK ERROR SET

004522  074336                    EMT300
004524  074400                    EHT1
004526  074474                    EDT1
004530  074522                    EFT1


                         ;ERROR  301      SEARCH NOT ENABLED DURING DATA COMMAND

004532  074362                    EMT301
004534  074400                    EHT1
004536  074474                    EDT1
004540  074522                    EFT1


                         ;PUT ERROR TABLE HERE

```

```
                                        .SBTTL  START OF PROGRAM


        004542                          START:

                                        ;CLEAR AND SETUP FOR TEST EXECUTION
        004542  000240                          NOP
        004544  000005                          RESET                           ;INITIALIZE THE SYSTEM

                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
        004546  012706  001114                  MOV     #$CMTAG,R6              ;;FIRST LOCATION TO BE CLEARED
        004552  005026                          CLR     (R6)+                   ;;CLEAR MEMORY LOCATION
        004554  022706  001154                  CMP     #SWR,R6 ;;DONE?
        004560  001374                          BNE     .-6                     ;;LOOP BACK IF NO
        004562  012706  001100                  MOV     #STACK,SP               ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
        004566  012737  062064  000020          MOV     #$SCOPE,@#IOTVEC        ;;IOT VECTOR FOR SCOPE ROUTINE
        004574  012737  000340  000022          MOV     #340,@#IOTVEC+2         ;;LEVEL 7
        004602  012737  062640  000030          MOV     #$ERROR,@#EMTVEC        ;;EMT VECTOR FOR ERROR ROUTINE
        004610  012737  000340  000032          MOV     #340,@#EMTVEC+2         ;;LEVEL 7
        004616  012737  064400  000034          MOV     #$TRAP,@#TRAPVEC        ;;TRAP VECTOR FOR TRAP CALLS
        004624  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
        004632  012737  064506  000024          MOV     #$PWRDN,@#PWRVEC        ;;POWER FAILURE VECTOR
        004640  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;;LEVEL 7
        004646  013737  057140  057132          MOV     $ENDCT,$EOPCT           ;;SETUP END-OF-PROGRAM COUNTER
        004654  005037  001206                  CLR     $TIMES                  ;;INITIALIZE NUMBER OF ITERATIONS
        004660  005037  001210                  CLR     $ESCAPE                 ;;CLEAR THE ESCAPE ON ERROR ADDRESS
        004664  112737  000001  001131          MOVB    #1,$ERMAX               ;;ALLOW ONE ERROR PER TEST
        004672  012737  004672  001122          MOV     #.,$LPADR               ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
        004700  012737  004700  001124          MOV     #.,$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.
        004706  013746  000004                  MOV     @#ERRVEC,-(SP)          ;;SAVE ERROR VECTOR
        004712  012737  004746  000004          MOV     #64$,@#ERRVEC           ;;SET UP ERROR VECTOR
        004720  012737  177570  001154          MOV     #DSWR,SWR               ;;SETUP FOR A HARDWARE SWICH REGISTER
        004726  012737  177570  001156          MOV     #DDISP,DISPLAY          ;;AND A HARDWARE DISPLAY REGISTER
        004734  022777  177777  174212          CMP     #-1,@SWR                ;;TRY TO REFERENCE HARDWARE SWR
        004742  001012                          BNE     66$                     ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
        004744  000403                          BR      65$                     ;;BRANCH IF NO TIMEOUT
        004746  012716  004754          64$:    MOV     #65$,(SP)               ;;SET UP FOR TRAP RETURN
        004752  000002                          RTI
        004754  012737  000176  001154  65$:    MOV     #SWREG,SWR              ;;POINT TO SOFTWARE SWR
        004762  012737  000174  001156          MOV     #DISPREG,DISPLAY
        004770  012637  000004          66$:    MOV     (SP)+,@#ERRVEC          ;;RESTORE ERROR VECTOR

        004774  005037  001230                  CLR     $PASS                   ;;CLEAR PASS COUNT
        005000  132737  000200  001243          BITB    #APTSIZE,$ENVM          ;;TEST USER SIZE UNDER APT
        005006  001403                          BEQ     67$                     ;;YES,USE NON-APT SWITCH
        005010  012737  001244  001154          MOV     #$SWREG,SWR             ;;NO,USE APT SWITCH REGISTER
        005016                          67$:
        005016  012746  000140                  MOV     #PR3,-(SP)              ;;PUT NEW PS ON STACK
        005022  012746  005030                  MOV     #68$,-(SP)              ;;PUT NEW PC ON  STACK
        005026  000002                          RTI                             ;;POP NEW PC AND PS
        005030                          68$:
```

```
                                        .SBTTL  TYPE PROGRAM NAME
                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
005030  005227  177777                          INC     #-1                     ;;FIRST TIME?
005034  001060                                  BNE     69$                     ;;BRANCH IF NO
005036  022737  057274  000042                  CMP     #$ENDAD,@#42            ;;ACT-11?
005044  001454                                  BEQ     69$                     ;;BRANCH IF YES
005046  104401  005114                          TYPE    ,70$                    ;;TYPE ASCIZ STRING
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
005052  005737  000042                          TST     @#42                    ;;ARE WE RUNNING UNDER XXDP/ACT?
005056  001012                                  BNE     71$                     ;;BRANCH IF YES
005060  123727  001242  000001                  CMPB    $ENV,#1                 ;;ARE WE RUNNING UNDER APT?
005066  001406                                  BEQ     71$                     ;;BRANCH IF YES
005070  023727  001154  000176                  CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
005076  001005                                  BNE     72$                     ;;BRANCH IF NO
005100  104407                                  GTSWR                           ;;GET SOFT-SWR SETTINGS
005102  000403                                  BR      72$
005104  112737  000001  001150  71$:    MOVB    #1,$AUTOB               ;;SET AUTO-MODE INDICATOR
005112                                  72$:
005112  000431                                  BR      69$                     ;;GET OVER THE ASCIZ
                                        ;;70$:  .ASCIZ  <CRLF>@MD-11-DZRMJ-A, RM03 DISKLESS DIAGNOSTIC PROGRAM@<CRLF>
005176                                  69$:


                                        ;FIND OUT IF PROGRAM IS RUNNING IN STANDALONE MODE
005176  005737  000042                          TST     42                      ;IS LOC 42 ZERO ??
005202  001003                                  BNE     10$                     ;NO - NOT IN STANDALONE
005204  105737  001242                          TSTB    $ENV                    ;IS APT ENVIRONMENT ZERO ??
005210  001411                                  BEQ     STANDALONE              ;YES - PROGRAM IN STANDALONE
005212                                  10$:

                                        ;PROGRAM NOT RUNNING IN STANDALONE - SEE IF SIZING IS ALLOWED
005212  132737  000200  001243                  BITB    #BIT7,$ENVM             ;SIZING ALLOWED ??
005220  001003                                  BNE     20$                     ;NO
005222  012737  000377  001300                  MOV     #377,$DEVM              ;YES - SET DEVICE MAP FOR ALL DEVICES
005230                                  20$:

                                        ;GO TO COMMON START CODE
005230  000137  006052                          JMP     CMNSTART
```

```
005234                            STANDALONE:
005234  004737  063050                    JSR     PC,$TKINT
                                  ;SEE IF THIS IS THE FIRST START AFTER PROGRAM WAS LOADED
005240  005327  000001                    DEC     #1              ;FIRST START ??
005244  100024                            BPL     10$             ;YES !!
                                  ;SEE IF THE USER WANTS TO KEEP SAME DEVICES FOR TESTING
005246  104401  065272            5$:     TYPE    ,CNSL00         ;MAINTAIN PREVIOUS PARAMETERS??
005252  104411                            RDCHR                   ;GET RESPONSE
005254  012637  001176                    MOV     (SP)+,$TMP1     ;ECHO RESPONSE
005260  104401  001176                    TYPE    ,$TMP1
005264  123727  001176  000131            CMPB    $TMP1,#'Y       ;YES RESPONSE??
005272  001002                            BNE     6$              ;NO!!
005274  000137  006212                    JMP     READY           ;KEEP PREVIOUS PARAMETERS
005300  123727  001176  000116    6$:     CMPB    $TMP1,#'N       ;NO RESPONSE??
005306  001420                            BEQ     20$             ;GET NEW PARAMETERS
005310  104401  065141                    TYPE    ,QSTMRK         ;NOT YES OR NO, TYPE "?"
005314  000754                            BR      5$              ;RETRY
005316                            10$:
                                  ;SEE IF OPERATOR WANTS HELP FILE
005316  104401  065143                    TYPE    ,HELPQST        ;WANT HELP ??
005322  104411                            RDCHR                   ;GET RESPONSE
005324  012637  001176                    MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
005330  104401  001176                    TYPE    ,$TMP1
005334  123727  001176  000131            CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ??
005342  001002                            BNE     20$             ;NO - DONT TYPE HELP
005344  104401  106652                    TYPE    ,HELP           ;YES - TYPE HELP TEXT
005350                            20$:
                                  ;SEE IF USER WANTS TO CHANGE RM03 UNIBUS ADDRESS
005350  104401  065177                    TYPE    ,UBUSQST        ;WANT TO CHANGE ADDRESS ??
005354  104411                            RDCHR                   ;GET RESPONSE
005356  012637  001176                    MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
005362  104401  001176                    TYPE    ,$TMP1
005366  123727  001176  000131            CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ??
005374  001137                            BNE     90$             ;NO !!
005376                            30$:
                                  ;DIALOGUE TO CHANGE THE UNIBUS ADDRESS, AND INTERRUPT VECTOR
005376  104401  065331                    TYPE    ,CNSL01         ;TYPE CURRENT BUS ADDRESS
005402  013746  001276                    MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
005406  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
005410  104401  065132                    TYPE    ,CLSPRN
005414  104413                            RDOCT                   ;GET NEW BUS ADDRESS
005416  012637  001176                    MOV     (SP)+,$TMP1     ;CARRIAGE RETURN??
005422  001412                            BEQ     50$             ;YES-SKIP TO NEXT ENTRY
005424  022737  160000  001176            CMP     #160000,$TMP1   ;BASE ADDRESS IN I/O PAGE??
005432  101403                            BLOS    40$             ;YES
005434  104401  065356                    TYPE    ,CNSL02         ;TYPE WARNING MESSAGE
005440  000756                            BR      30$             ;RETRY
005442  013737  001176  001276    40$:    MOV     $TMP1,$BASE     ;STORE NEW BUS ADDRESS
005450  113737  001272  001176    50$:    MOVB    $VECT1,$TMP1    ;TYPE CURRENT VECTOR ADDRESS
005456  105037  001177                    CLRB    $TMP1+1
```

```
005462  104401  065437                  TYPE    ,CNSL03
005466  013746  001176                  MOV     $TMP1,-(SP)             ;;SAVE $TMP1 FOR TYPEOUT
005472  104403                          TYPOS                           ;;GO TYPE--OCTAL ASCII
005474     003                          .BYTE   3                       ;;TYPE 3 DIGIT(S)
005475     000                          .BYTE   0                       ;;SUPPRESS LEADING ZEROS
005476  104401  065132                  TYPE    ,CLSPRN
005502  104413                          RDOCT                           ;GET NEW VECTOR ADDRESS
005504  012637  001176                  MOV     (SP)+,$TMP1             ;CARRIAGE RETURN?
005510  001412                          BEQ     70$                     ;YES-SKIP TO NEXT ENTRY
005512  022737  001000  001176          CMP     #1000,$TMP1             ;VECTOR ADDRESS < 1000??
005520  101003                          BHI     60$                     ;YES!!
005522  104401  065467                  TYPE    ,CNSL04                 ;TYPE WARNING MESSAGE
005526  000750                          BR      50$                     ;RETRY
005530  113737  001176  001272  60$:    MOVB    $TMP1,$VECT1            ;STORE NEW VECTOR ADDRESS
005536  113737  001273  001176  70$:    MOVB    $VECT1+1,$TMP1          ;TYPE CURRENT PRIORITY
005544  006237  001176                  ASR     $TMP1
005550  006237  001176                  ASR     $TMP1
005554  006237  001176                  ASR     $TMP1
005560  006237  001176                  ASR     $TMP1
005564  006237  001176                  ASR     $TMP1
005570  105037  001177                  CLRB    $TMP1+1
005574  104401  065543                  TYPE    ,CNSL05
005600  013746  001176                  MOV     $TMP1,-(SP)             ;;SAVE $TMP1 FOR TYPEOUT
005604  104403                          TYPOS                           ;;GO TYPE--OCTAL ASCII
005606     001                          .BYTE   1                       ;;TYPE 1 DIGIT(S)
005607     000                          .BYTE   0                       ;;SUPPRESS LEADING ZEROS
005610  104401  065132                  TYPE    ,CLSPRN
005614  104413                          RDOCT                           ;GET NEW PRIORITY
005616  012637  001176                  MOV     (SP)+,$TMP1             ;CARRIAGE RETURN??
005622  001424                          BEQ     90$                     ;YES-SKIP TO NEXT ENTRY
005624  023727  001176  000007          CMP     $TMP1,#7                ;LEGAL PRIORITY??
005632  002403                          BLT     80$                     ;YES!!
005634  104401  065577                  TYPE    ,CNSL06                 ;TYPE WARNING MESSAGE
005640  000736                          BR      70$                     ;RETRY
005642                          80$:                                    ;STORE NEW PRIORITY
005642  006337  001176                  ASL     $TMP1
005646  006337  001176                  ASL     $TMP1
005652  006337  001176                  ASL     $TMP1
005656  006337  001176                  ASL     $TMP1
005662  006337  001176                  ASL     $TMP1
005666  113737  001176  001273          MOVB    $TMP1,$VECT1+1
005674                          90$:

                                ;DIALOGUE TO INPUT DEVICE NUMBERS
005674  005037  001300                  CLR     $DEVM                   ;CLEAR DEVICE MAP
005700  104401  065624                  TYPE    ,CNSL07                 ;TYPE INPUT INSTRUCTIONS
005704  104401  065135                  TYPE    ,PROMPT                 ;TYPE PROMPTING CHARACTER
005710  104411                          RDCHR                           ;GET RESPONSE
005712  012637  001176                  MOV     (SP)+,$TMP1             ;ECHO RESPONSE
005716  104401  001176                  TYPE    ,$TMP1
005722  023727  001176  000101          CMP     $TMP1,#'A               ;TEST ALL DRIVES??
005730  001017                          BNE     110$                    ;NO
005732  012737  000377  001300          MOV     #377,$DEVM              ;TEST ALL DEVICES
005740  000444                          BR      140$                    ;SKIP TO NEXT ENTRY
005742  104401  065135          100$:   TYPE    ,PROMPT                 ;TYPE PROMPTING CHARACTER
005746  104411                          RDCHR                           ;GET RESPONSE
```

```
005750  012637  001176                   MOV     (SP)+,$TMP1       ;ECHO RESPONSE
005754  104401  001176                   TYPE    ,$TMP1
005760  023727  001176  000015           CMP     $TMP1,#CR         ;CARRIAGE RETURN??
005766  001431                           BEQ     140$
005770  023727  001176  000060   110$:   CMP     $TMP1,#'0         ;NUMBER < 0??
005776  002404                           BLT     120$              ;YES!!
006000  023727  001176  000067           CMP     $TMP1,#'7         ;NUMBER > 7??
006006  003403                           BLE     130$              ;NO!!
006010  104401  065141           120$:   TYPE    ,QSTMRK           ;TYPE "?"
006014  000752                           BR      100$              ;RETRY
006016  013701  001176           130$:   MOV     $TMP1,R1          ;R1=DRIVE NUMBER
006022  042701  177770                   BIC     #^C7,R1
006026  116102  066064                   MOVB    ATNTBL(R1),R2               ;DECODE DEVICE NUMBER
006032  042702  177400                   BIC     #^C377,R2                   ;CLEAR UNUSED BITS
006036  050237  001300                   BIS     R2,$DEVM                    ;SET DEVICE # IN MAP
006042  122737  000377  001300           CMPB    #377,$DEVM        ;DONE ??
006050  101334                           BHI     100$              ;NO
006052                           140$:
```

```
006052                              CMNSTART:

                                    ;ASSEMBLE TEST QUE FROM DEVICE MAP
006052  013700  001300                        MOV     $DEVM,R0           ;R0 = DEVICE MAP
006056  012701  001460                        MOV     #TSTQUE+2,R1       ;R1 = ADDRESS OF FIRST ENTRY IN QUE
006062  010137  001456                        MOV     R1,TSTQUE          ;INITIALIZE ENTRY POINTER
006066  012702  000001                        MOV     #1,R2              ;R2 = DEVICE POINTER
006072  005003                                CLR     R3                 ;R3 = DEVICE NUMBER
006074  030200                      10$:      BIT     R2,R0              ;IS THIS DEVICE IN MAP ??
006076  001406                                BEQ     20$                ;NO !!
006100  010311                                MOV     R3,(R1)            ;YES - ENTER DEVICE NUMBER IN QUE
006102  116361  066064  000001                MOVB    ATNTBL(R3),1(R1);ENTER ATTENTION BIT IN QUE
006110  062701  000002                        ADD     #2,R1              ;ADVANCE ENTRY POINTER
006114  006302                      20$:      ASL     R2                 ;ADVANCE DEVICE POINTER
006116  105702                                TSTB    R2                 ;DONE ALL DEVICES ??
006120  001402                                BEQ     25$                ;YES
006122  005203                                INC     R3                 ;ADVANCE DEVICE NUMBER
006124  000763                                BR      10$                ;ENTER NEXT DEVICE
006126  005011                      25$:      CLR     (R1)               ;TERMINATE TEST QUE

                                    ;SIZE FOR CLOCK
006130  004737  060116                        JSR     PC,SIZCLK          ;SEE IF CLOCK PRESENT
006134  000403                                BR      40$                ;YES - CLOCK IS PRESENT
006136  104000                      30$:      ERROR                      ;NO CLOCK
006140  000000                                HALT
006142  000775                                BR      30$
006144                              40$:
                                    .SBTTL    GET VALUE FOR SOFTWARE SWITCH REGISTER
006144  005737  000042                        TST     @#42               ;;ARE WE RUNNING UNDER XXDP/ACT?
006150  001012                                BNE     64$                ;;BRANCH IF YES
006152  123727  001242  000001                CMPB    $ENV,#1            ;;ARE WE RUNNING UNDER APT?
006160  001406                                BEQ     64$                ;;BRANCH IF YES
006162  023727  001154  000176                CMP     SWR,#SWREG         ;;SOFTWARE SWITCH REG SELECTED?
006170  001005                                BNE     65$                ;;BRANCH IF NO
006172  104407                                GTSWR                      ;;GET SOFT-SWR SETTINGS
006174  000403                                BR      65$
006176  112737  000001  001150      64$:      MOVB    #1,$AUTOB          ;;SET AUTO-MODE INDICATOR
006204                              65$:
006204  012737  000140  000032                MOV     #PR3,@#EMTVEC+2 ;DROP PRIORITY DURING ERROR TYPEOUT
006212  000240                      READY:    NOP                        ;READY TO START TEST
006214  004737  063050                        JSR     PC,$TKINT          ;INITIALIZE TTY
006220  117737  173232  001234                MOVB    @TSTQUE,$UNIT      ;LOAD UNIT NUMBER
```

```
        .SBTTL  REGISTER AND STORAGE USAGE

        ;REGISTER ASSIGNMENTS
        ;
        ;R0      =       UNIBUS ADDRESS OF RH CONTROLLER
        ;R1      =       ADDRESS OF ENTRY IN TEST QUE CORRESPONDING TO THE
        ;                UNIT UNDER TEST
        ;R2,R3   =       WORKING REGISTERS FOR TEST IN PROGRESS, MUST BE
        ;                SAVED BY SUBROUTINES
        ;R4,R5   =       GENERAL WORKING REGISTERS, ARE NOT SAVED BY
        ;                SUBROUTINES
        ;R6      =       STACK POINTER
        ;R7      =       LINKAGE REGISTER TO SUBROUTINES

        ;STORAGE ASSIGNMENTS
        ;
        ;$TMP0-$TMP4     TEMPORARY STORAGE, NOT SAVED BY SUBROUTINES
        ;$GDDAT,$BDDAT   EXPECTED AND RRECEIVED STATUS FOR ERROR TYPEOUT
        ;$GDADR,$BDADR   ADDRESS OF EXPECTED AND RECEIVED STATUS IF APPLICABLE,
        ;                ALSO THE ADDRESS OF A REGISTER ERROR
        ;
        ;$TSTN   =       TEST NUMBER
        ;$UNIT   =       NUMBER OF DEVICE BEING TESTED
        ;RGINBF  =       THE REGISTER INPUT BUFFER HAS A STORAGE LOCATION FOR
        ;                EACH REGISTER, AND IS USED WHEN READING STATUS AND
        ;                CONTROL DATA
        ;RGOTBF  =       THE REGISTER OUTPUT BUFFER HAS A STORAGE LOCATION FOR
        ;                EACH REGISTER, AND IS USED FOR ASSEMBLING DATA TO BE
        ;                WRITTEN IN REGISTERS
```

```
                                        ;;*****************************************************************
                                        ;*TEST 1          TRANSFER TEST

                                        ;;*****************************************************************
006226  000004                          TST1:   SCOPE
006230  012737  000001  001226                  MOV     #1,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX

006236  000240                                  NOP
006240  012737  000024  001120                  MOV     #20.,$ICNT              ;20 ITERATIONS
006246  112737  000001  001131                  MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
006254  012737  006270  001122                  MOV     #T1,$LPADR      ;LOAD LOOP ON TEST ADDRESS
006262  012737  006270  001124                  MOV     #T1,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
006270                                  T1:
006270  012706  001100                          MOV     #STACK,SP               ;LOAD THE STACK POINTER
006274  013700  001276                          MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
006300  013701  001456                          MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
006304  012702  000000                          MOV     #0,R2                   ;R2 = REGISTER INDEX
006310                                  10$:
                                        ;CLEAR THE MASSBUS AND VERIFY THAT NONEXISTANT DEVICE ERROR IS RESET
006310  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
006316  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

006322  016037  000010  001142                  MOV     RMCS2(R0),$BDDAT        ;STORE RMCS2 AT $BDDAT
006330  032737  010000  001142                  BIT     #NED,$BDDAT
006336  001417                                  BEQ     20$
006340  111137  001140                          MOVB    (R1),$GDDAT
006344  042737  177770  001140                  BIC     #↑CUNTMSK,$GDDAT
006352  052737  000100  001140                  BIS     #IR,$GDDAT
006360  010037  001136                          MOV     R0,$BDADR
006364  062737  000010  001136                  ADD     #RMCS2,$BDADR
006372  104001                                  ERROR   1
006374  000500                                  BR      60$
006376                                  20$:
                                        ;READ THE REGISTER WHOSE INDEX IS IN R2 AND EXIT TEST IF THE READ
                                        ;DOES NOT SET "NED" ERROR
006376  010003                                  MOV     R0,R3                   ;R3 = REGISTER ADDRESS
006400  060203                                  ADD     R2,R3
006402  011304                                  MOV     (R3), R4                ;READ REGISTER
006404  032760  010000  000010                  BIT     #NED, RMCS2(R0)         ;IS "NED" SET??
006412  001473                                  BEQ     70$                     ;NO!!
006414  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
006422  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

006426  016037  000010  001142                  MOV     RMCS2(R0),$BDDAT        ;STORE RMCS2 AT $BDDAT
006434  032737  010000  001142                  BIT     #NED,$BDDAT
006442  001417                                  BEQ     30$
006444  111137  001140                          MOVB    (R1),$GDDAT
006450  042737  177770  001140                  BIC     #↑CUNTMSK,$GDDAT
006456  052737  000100  001140                  BIS     #IR,$GDDAT
006464  010037  001136                          MOV     R0,$BDADR
006470  062737  000010  001136                  ADD     #RMCS2,$BDADR
006476  104001                                  ERROR   1
006500  000436                                  BR      60$
006502                                  30$:
                                        ;WRITE THE REGISTER WHOSE INDEX IS IN R2 AND EXIT TEST IF THE WRITE
                                        ;DOES NOT SET "NED" ERROR
```

```
006502  012713  000000                  MOV     #0,(R3)         ;WRITE REGISTER
006506  032760  010000  000010          BIT     #NED, RMCS2     (R0)    ;IS "NED" SET??
006514  001432                          BEQ     70$             ;NO!!

006516                          40$:
                                ;COULD NOT READ OR WRITE THE REGISTER WITHOUT SETING "NED" ERROR -
                                ;ADVANCE THE REGISTER INDEX AND REPEAT THE TEST FOR THE NEXT
                                ;AVAILABLE DEVICE REGISTER
006516  062702  000002                  ADD     #2,R2           ;ADVANCE TO NEXT REGISTER
006522  022702  000002                  CMP     #RMWC,R2                ;IS THIS RMWC??
006526  001773                          BEQ     40$             ;YES - TRY NEXT REGISTER
006530  022702  000004                  CMP     #RMBA,R2                ;IS THIS RMBA??
006534  001770                          BEQ     40$             ;YES - TRY NEXT REGISTER
006536  022702  000010                  CMP     #RMCS2,R2               ;IS THIS RMCS5??
006542  001765                          BEQ     40$             ;YES - TRY ANOTHER REGISTER
006544  022702  000016                  CMP     #RMAS,R2                ;IS THIS RMAS ??
006550  001762                          BEQ     40$             ;YES - TRY ANOTHER REGISTER
006552  022702  000022                  CMP     #RMDB,R2                ;IS THIS RMDB??
006556  001757                          BEQ     40$             ;YES - TRY ANOTHER REGISTER
006560  022702  000046                  CMP     #RMEC2,R2               ;IS THIS A LEGAL REGISTER
006564  103251                          BHIS    10$             ;YES - TRY THIS REGISTER

006566                          50$:

                                ;GOT 'NONEXISTENT DEVICE' ERROR FOR EVERY REMOTE REGISTER ADDRESS
006566  013737  001276  001136          MOV     $BASE, $BDADR           ;STORE BASE ADDRESS
006574  104002                          ERROR   2               ;DEMAND OR TRANSFER FAILED
006576  000137  057050          60$:    JMP     $EOSP           ;GO SELECT NEXT DEVICE

006602                          70$:

                                ;;*****************************************************************
                                ;*TEST 2         CTOD TEST

                                ;;*****************************************************************
006602  000004                  TST2:   SCOPE
006604  012737  000002  001226          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
006612  000240                          NOP
006614  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
006622  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
006630  012737  006644  001122          MOV     #T2,$LPADR      ;LOAD LOOP ON TEST ADDRESS
006636  012737  006644  001124          MOV     #T2,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
006644                          T2:
006644  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
006650  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
006654  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
006660  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
006666  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                ;WRITE ONES IN REMOTE REGISTERS

006672  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1

006700  012760  177777  000006          MOV     #-1,RMDA(R0)    ;LOAD RMDA

006706  012760  001777  000034          MOV     #CYLMSK,RMDC(R0)        ;LOAD RMDC

```

```
006714  012760  016200  000032          MOV     #FMT16!ECI!HCI!OFD,RMOF(R0)     ;LOAD RMOF
                                ;READ REMOTE REGISTERS TWICE
006722  012702  000001                  MOV     #1,R2
006726                          10$:

006726  016037  000000  001326          MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER

006734  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER

006742  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER

006750  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
006756  005302                          DEC     R2
006760  100362                          BPL     10$
                                ;SEE IF ANY ONE BITS CAME BACK
006762  042737  177701  001326          BIC     #^CILF76,RMCS1I ;IS RMCS1 0??
006770  001014                          BNE     20$             ;NO!!
006772  005737  001334                  TST     RMDAI           ;IS RMDA 0??
006776  001011                          BNE     20$             ;NO!!
007000  042737  176000  001362          BIC     #XNUDC,RMDCI    ;IS RMDC 0??
007006  001005                          BNE     20$             ;NO!!
007010  042737  161577  001360          BIC     #XNUOF,RMOFI    ;IS RMOF 0 ??
007016  001001                          BNE     20$             ;NO!!
                                ;CANNOT READ ANY ONE BITS FROM REMOTE REGISTERS
007020  104003                          ERROR   3               ;CTOD MUST BE STUCK
007022                          20$:


                                ;;*****************************************************************
                                ;*TEST 3        MASSBUS INITIALIZE TEST

                                ;;*****************************************************************
007022  000004                  TST3:   SCOPE
007024  012737  000003  001226          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

007032  000240                          NOP
007034  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
007042  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
007050  012737  007064  001122          MOV     #T3,$LPADR  ;LOAD LOOP ON TEST ADDRESS
007056  012737  007064  001124          MOV     #T3,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
007064                          T3:
007064  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
007070  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
007074  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
007100  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
007106  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                ;WRITE ONES IN SELECTED REGISTERS

007112  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1

007120  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1

007126  012760  177777  000042          MOV     #-1,RMER2(R0)   ;LOAD RMER2
                                ;USING CONTROLLER CLEAR, IE., BIT 5 OF RMCS2, INITIALIZE THE MASSBUS
007134  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
007142  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                ;READ THE REGISTERS THAT WERE WRITTEN
```

```
007146 016037 000000 001326         MOV     RMCS1(R0),RMCS1I            ;STORE RMCS1 IN INPUT BUFFER
007154 016037 000014 001342         MOV     RMER1(R0),RMER1I            ;STORE RMER1 IN INPUT BUFFER
007162 016037 000042 001370         MOV     RMER2(R0),RMER2I            ;STORE RMER2 IN INPUT BUFFER
                            ;SEE IF ANY REGISTER BITS WERE CLEARED
007170 052737 177701 001326         BIS     #^CILF76,RMCS1I             ;SET ANY BIT NOT WRITTEN
007176 052737 001567 001370         BIS     #XNUER2,RMER2I
007204 022737 177777 001326         CMP     #-1,RMCS1I                  ;ANY ZEROS IN RMCS1??
007212 001011                       BNE     10$             ;YES!!
007214 022737 177777 001342         CMP     #-1,RMER1I                  ;ANY ZEROS IN RMERI??
007222 001005                       BNE     10$             ;YES!!
007224 022737 177777 001370         CMP     #-1,RMER2I                  ;ANY ZEROS IN RMER2??
007232 001001                       BNE     10$
                            ;NONE OF THE BITS WERE CLEARED
007234 104004                       ERROR   4               ;MASSBUS INIT FAILED
007236                      10$:

                            ;;*********************************************************
                            ;*TEST 4        CLEAR STUCK ACTIVE TEST

                            ;;*********************************************************
007236 000004               TST4:   SCOPE
007240 012737 000004 001226         MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

007246 000240                       NOP
007250 012737 000024 001120         MOV     #20.,$ICNT      ;20 ITERATIONS
007256 112737 000001 001131         MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
007264 012737 007300 001122         MOV     #T4,$LPADR  ;LOAD LOOP ON TEST ADDRESS
007272 012737 007300 001124         MOV     #T4,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
007300                      T4:
007300 012706 001100                MOV     #STACK,SP       ;LOAD THE STACK POINTER
007304 013700 001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
007310 013701 001456                MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
007314 012760 000040 000010         MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
007322 111160 000010                MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                            ;WRITE ONES IN TEST REGISTERS

007326 012760 177777 000014         MOV     #-1,RMER1(R0)   ;LOAD RMER1

007334 012760 177777 000042         MOV     #-1,RMER2(R0)   ;LOAD RMER2

007342 012760 000001 000024         MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
                            ;READ TEST REGISTERS AND SEE IF ANY BITS ARE ON

007350 016037 000014 001342         MOV     RMER1(R0),RMER1I            ;STORE RMER1 IN INPUT BUFFER

007356 016037 000042 001370         MOV     RMER2(R0),RMER2I            ;STORE RMER2 IN INPUT BUFFER

007364 016037 000024 001352         MOV     RMMR1(R0),RMMR1I            ;STORE RMMR1 IN INPUT BUFFER
007372 042737 040000 001342         BIC     #UNS,RMER1I     ;DONT ACCEPT UNSAFE
007400 001011                       BNE     10$             ;BRANCH IF ANY OTHER BITS ON
007402 042737 040200 001370         BIC     #SKI!DVC,RMER2I ;DONT ACCEPT SKI OR DVC
007410 001005                       BNE     10$             ;BRANCH IF ANY OTHER BITS ON
007412 032737 000001 001352         BIT     #DMD,RMMR1I     ;BRANCH IF DMD IS ON
```

```
007420  001001                               BNE     10$
007422  104026                               ERROR   26                ;MBA CLR IS STUCK ACTIVE
007424                               10$:


                                     ;;*****************************************************************
                                     ;*TEST 5          TRISTATE TRANSFER TEST

                                     ;;*****************************************************************
007424  000004                       TST5:   SCOPE
007426  012737  000005  001226               MOV     #5,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX
007434  000240                               NOP
007436  012737  000024  001120               MOV     #20.,$ICNT        ;20 ITERATIONS
007444  112737  000001  001131               MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
007452  012737  007466  001122               MOV     #T5,$LPADR   ;LOAD LOOP ON TEST ADDRESS
007460  012737  007466  001124               MOV     #T5,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
007466                               T5:
007466  012706  001100                       MOV     #STACK,SP         ;LOAD THE STACK POINTER
007472  013700  001276                       MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
007476  013701  001456                       MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
007502  005002                               CLR     R2                ;CLEAR ERROR FLAGS
007504  012760  000040  000010               MOV     #CLR,RMCS2(R0)    ;CLEAR THE MASSBUS
007512  111160  000010                       MOVB    (R1),RMCS2(R0)    ;SELECT UNIT
                                     ;WRITE ONES IN SELECTED REGISTERS

007516  012760  000076  000000               MOV     #ILF76,RMCS1(R0)          ;LOAD RMCS1

007524  012760  177777  000006               MOV     #-1,RMDA(R0)      ;LOAD RMDA

007532  012760  177777  000014               MOV     #-1,RMER1(R0)     ;LOAD RMER1

007540  012760  177777  000032               MOV     #-1,RMOF(R0)      ;LOAD RMOF

007546  012760  177777  000042               MOV     #-1,RMER2(R0)     ;LOAD RMER2
                                     ;WRITE ZEROS IN SELECTED REGISTERS

007554  012760  000000  000000               MOV     #0,RMCS1(R0)      ;LOAD RMCS1

007562  012760  000000  000006               MOV     #0,RMDA(R0)       ;LOAD RMDA

007570  012760  000000  000014               MOV     #0,RMER1(R0)      ;LOAD RMER1

007576  012760  000000  000032               MOV     #0,RMOF(R0)       ;LOAD RMOF

007604  012760  000000  000034               MOV     #0,RMDC(R0)       ;LOAD RMDC

007612  012760  000000  000042               MOV     #0,RMER2(R0)      ;LOAD RMER2
                                     ;READ BACK ALL REGISTERS

007620  016037  000000  001326               MOV     RMCS1(R0),RMCS1I          ;STORE RMCS1 IN INPUT BUFFER

007626  016037  000006  001334               MOV     RMDA(R0),RMDAI   ;STORE RMDA IN INPUT BUFFER

007634  016037  000014  001342               MOV     RMER1(R0),RMER1I          ;STORE RMER1 IN INPUT BUFFER

007642  016037  000032  001360               MOV     RMOF(R0),RMOFI   ;STORE RMOF IN INPUT BUFFER
```

```

007650  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER

007656  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
                                ;CHECK EACH REGISTER CONTENT FOR ZERO BITS WRITTEN & READ
007664  012702  177777                  MOV     #-1,R2          ;ACCUMULATE ZEROS IN R2
007670  052737  177701  001326          BIS     #↑C<ILF76>,RMCS1I ;SET ALL BITS NOT WRITTEN
007676  052737  161577  001360          BIS     #XNUOF,RMOFI
007704  052737  176000  001362          BIS     #XNUDC,RMDCI
007712  052737  001567  001370          BIS     #XNUER2,RMER2I
007720  005137  001326                  COM     RMCS1I          ;COMPLEMENT REGISTER CONTENTS
007724  005137  001334                  COM     RMDAI
007730  005137  001342                  COM     RMER1I
007734  005137  001360                  COM     RMOFI
007740  005137  001362                  COM     RMDCI
007744  005137  001370                  COM     RMER2I
007750  043702  001326                  BIC     RMCS1I,R2               ;ACCUMULATE ALL ZERO BITS
007754  043702  001334                  BIC     RMDAI,R2
007760  043702  001342                  BIC     RMER1I,R2
007764  043702  001360                  BIC     RMOFI,R2
007770  043702  001362                  BIC     RMDCI,R2
007774  043702  001370                  BIC     RMER2I,R2
010000  001407                          BEQ     10$             ;BRANCH IF EACH BIT IS ZERO
                                ;ONE OR MORE BIT POSITIONS ARE NOT ZERO
010002  010237  001142                  MOV     R2,$BDDAT               ;SAVE RESULT FOR TYPE
010006  005037  001140                  CLR     $GDDAT          ;LOAD EXPECTED RESULT
010012  104005                          ERROR   5               ;TRISTATE BUS IS STUCK AT ONE
010014  052702  000001                  BIS     #BIT0,R2        ;SET ERROR FLAG
010020                          10$:

010020  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
010026  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                ;PRESET SELECTED REGISTERS TO ZEROS
                                ;(ASSUME RMCS1, RMER1, RMER2 WERE CLEARED BY INIT)

010032  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA

010040  012760  000000  000032          MOV     #0,RMOF(R0)     ;LOAD RMOF

010046  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC
                                ;WRITE ONES IN SELECTED REGISTERS

010054  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1

010062  012760  177777  000006          MOV     #-1,RMDA(R0)    ;LOAD RMDA

010070  012760  016200  000032          MOV     #↑CXNUOF,RMOF(R0)       ;LOAD RMOF

010076  012760  001777  000034          MOV     #↑CXNUDC,RMDC(R0)       ;LOAD RMDC

010104  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1

010112  012760  176210  000042          MOV     #↑CXNUER2,RMER2(R0)     ;LOAD RMER2
                                ;READ ALL REGISTERS

010120  016037  000000  001326          MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
```

```
010126 016037 000006 001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
010134 016037 000032 001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
010142 016037 000034 001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
010150 016037 000014 001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
010156 016037 000042 001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
                             ;CHECK EACH REGISTER CONTENT FOR ONE BITS WRITTEN & READ
010164 042737 177701 001326          BIC     #↑CILF76,RMCS1I         ;CLEAR ALL BITS NOT WRITTEN
010172 042737 161577 001360          BIC     #XNUOF,RMOFI
010200 042737 176000 001362          BIC     #XNUDC,RMDCI
010206 042737 001567 001370          BIC     #XNUER2,RMER2I
010214 005002                        CLR     R2              ;ACCUMULATE ONES IN R2
010216 053702 001326                 BIS     RMCS1I,R2               ;ACCUMULATE ALL ONE BITS
010222 053702 001334                 BIS     RMDAI,R2
010226 053702 001360                 BIS     RMOFI,R2
010232 053702 001362                 BIS     RMDCI,R2
010236 053702 001342                 BIS     RMER1I,R2
010242 053702 001370                 BIS     RMER2I,R2
010246 022702 177777                 CMP     #-1,R2          ;SEE IF EACH BIT POSITION WAS ONE
010252 001410                        BEQ     20$             ;BRANCH IF NONE STUCK
                             ;ONE OR MORE BIT POSITIONS ARE NOT ONE
010254 010237 001142                 MOV     R2,$BDDAT               ;SAVE RESULT FOR TYPE
010260 012737 177777 001140          MOV     #-1,$GDDAT              ;EXPECTED RESULT
010266 104006                        ERROR   6
010270 052702 000002                 BIS     #BIT1,R2        ;SET ERROR FLAG
010274                       20$:
010274 005702                        TST     R2              ;ANY ERRORS DETECTED ??
010276 001131                        BNE     30$             ;YES - DONT DO BIT TEST
010300 012702 000001                 MOV     #1,R2           ;R2=BIT POSITION
010304                       25$:
010304 012760 000040 000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
010312 111160 000010                 MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                             ;WRITE THE BIT PATTERN IN SELECTED DEVICE REGISTERS
010316 010260 000006                 MOV     R2,RMDA(R0)     ;LOAD RMDA
010322 010260 000032                 MOV     R2,RMOF(R0)     ;LOAD RMOF
010326 010260 000034                 MOV     R2,RMDC(R0)     ;LOAD RMDC
010332 010260 000014                 MOV     R2,RMER1(R0)    ;LOAD RMER1
010336 010260 000042                 MOV     R2,RMER2(R0)    ;LOAD RMER2
                             ;READ BACK THE REGISTERS
010342 016037 000006 001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
010350 016037 000032 001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
010356 016037 000034 001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
```

```
010364  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER

010372  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
                                ;CHECK REGISTER CONTENTS FOR CORRECT PATTERN
010400  005003                          CLR     R3                      ;R3=ACCUMULATED ONE BIT
010402  012704  177777                  MOV     #-1,R4                  ;R4=ACCUMULATED ZERO BITS
010406  013705  001334                  MOV     RMDAI,R5                ;GET ANY GOOD BITS FROM RMDA
010412  050503                          BIS     R5,R3
010414  005105                          COM     R5
010416  040504                          BIC     R5,R4
010420  013705  001360                  MOV     RMOFI,R5                ;GET GOOD BITS FROM RMOF
010424  042705  161577                  BIC     #XNUOF,R5
010430  050503                          BIS     R5,R3
010432  005105                          COM     R5
010434  042705  161577                  BIC     #XNUOF,R5
010440  040504                          BIC     R5,R4
010442  013705  001362                  MOV     RMDCI,R5                ;GET GOOD BITS FROM RMDC
010446  042705  176000                  BIC     #XNUDC,R5
010452  050503                          BIS     R5,R3
010454  005105                          COM     R5
010456  042705  176000                  BIC     #XNUDC,R5
010462  040504                          BIC     R5,R4
010464  013705  001342                  MOV     RMER1I,R5               ;GET GOOD BITS FROM RMER1
010470  050503                          BIS     R5,R3
010472  005105                          COM     R5
010474  040504                          BIC     R5,R4
010476  013705  001370                  MOV     RMER2I,R5               ;GET GOOD BITS FROM RMER2
010502  042705  001567                  BIC     #XNUER2,R5
010506  050503                          BIS     R5,R3
010510  005105                          COM     R5
010512  042705  001567                  BIC     #XNUER2,R5
010516  040504                          BIC     R5,R4
010520  010205                          MOV     R2,R5                   ;RESET ALL ONES IN R3 EXCEPT
010522  005105                          COM     R5                      ;FOR THE TEST BIT
010524  040503                          BIC     R5,R3
010526  040204                          BIC     R2,R4                   ;RESET TEST BIT IN R4
010530  050403                          BIS     R4,R3                   ;COMBINE ACCUMULATED 1'S + 0'S
010532  020302                          CMP     R3,R2                   ;IS PATTERN OK??
010534  001406                          BEQ     26$                     ;YES!!
010536  010237  001140                  MOV     R2,$GDDAT                       ;SAVE TEST PATTERN
010542  010337  001142                  MOV     R3,$BDDAT                       ;SAVE RESULT
010546  104007                          ERROR   7                       ;BIT INTERFERENCE IN TRISTATE BUS
010550  000404                          BR      30$                     ;SKIP TO NEXT
010552                          26$:
                                ;ADVANCE R2 TO THE NEXT PATTERN AND REPEAT TEST
010552  006302                          ASL     R2                      ;SHIFT THE BIT
010554  001402                          BEQ     30$                     ;EXIT IF DONE
010556  000137  010304                  JMP     25$
010562                          30$:

                                ;;*****************************************************************
                                ;*TEST 6         REGISTER SELECT TEST

                                ;;*****************************************************************
010562  000004                  TST6:   SCOPE
010564  012737  000006  001226          MOV     #6,$TESTN               ;;SET TEST NUMBER IN APT MAIL BOX
```

```

010572  000240                            NOP
010574  012737  000024  001120            MOV     #20.,$ICNT          ;20 ITERATIONS
010602  112737  000001  001131            MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
010610  012737  010624  001122            MOV     #T6,$LPADR    ;LOAD LOOP ON TEST ADDRESS
010616  012737  010624  001124            MOV     #T6,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
010624                            T6:
010624  012706  001100                    MOV     #STACK,SP           ;LOAD THE STACK POINTER
010630  013700  001276                    MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
010634  013701  001456                    MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
                                  ;THE FOLLOWING TABLE GIVES MASSBUS REGISTER SELECT VALUES FOR
                                  ;EACH DEVICE REGISTER
                                  ;
                                  ;       REGISTER        REG SEL
                                  ;       NAME            (16,8,4,2,1)
                                  ;
                                  ;       RMCS1           00000
                                  ;       RMDS            00001
                                  ;       RMER1           00010
                                  ;       RMMR1           00011
                                  ;       RMAS            00100
                                  ;       RMDA            00101
                                  ;       RMDT            00110
                                  ;       RMLA            00111
                                  ;       RMSN            01000
                                  ;       RMOF            01001
                                  ;       RMDC            01010
                                  ;       RMHR            01011
                                  ;       RMMR2           01100
                                  ;       RMER2           01101
                                  ;       RMEC1           01110
                                  ;       RMEC2           01111

                                  ;EACH REGISTER SELECT LINE IS TESTED FOR A STUCK AT ONE,
                                  ;STUCK AT ZERO FAULT. AS AN EXAMPLE, TO TEST REG SEL 1,
                                  ;FOR S-A-O, RMER1 IS WRITTEN WITH ZEROS. THEN THE REGISTER
                                  ;THAT HAS THE SAME SELECT VALUE, EXCEPT FOR THE SELECT LINE
                                  ;BEING TESTED, IS WRITTEN WITH ONES. IN THIS EXAMPLE,
                                  ;RMMR1 IS WRITTEN WITH ONES. IF SELECT LINE 1 IS S-A-O,
                                  ;THE ALL ONES WORD WILL BE WRITTEN IN RMER1, AND RMER1
                                  ;WILL NOT BE 0 WHEN READ BACK.

010640  005002                            CLR     R2                  ;R2= ZEROS SOURCE
010642  012703  177777                    MOV     #-1,R3              ;R3= ONES  SOURCE
                                  ;TEST REG SEL 1 FOR S-A-O
010646  012760  000040  000010            MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
010654  111160  000010                    MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

010660  010260  000014                    MOV     R2,RMER1(R0)        ;LOAD RMER1

010664  010260  000034                    MOV     R2,RMDC(R0)         ;LOAD RMDC

010670  010360  000024                    MOV     R3,RMMR1(R0)        ;LOAD RMMR1

010674  010360  000036                    MOV     R3,RMHR(R0)         ;LOAD RMHR

```

```
010700 016037 000014 001342              MOV     RMER1(R0),RMER1I         ;STORE RMER1 IN INPUT BUFFER

010706 016037 000034 001362              MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
010714 020337 001342                     CMP     R3,RMER1I
010720 001007                            BNE     10$
010722 052737 176000 001362              BIS     #XNUDC,RMDCI
010730 020337 001362                     CMP     R3,RMDCI
010734 001001                            BNE     10$
010736 104010                            ERROR   10              ;REG SEL 1 IS S-A-0
010740                           10$:

                                 ;TEST REG SEL 1 FOR S-A-1
010740 012760 000040 000010              MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
010746 111160 000010                     MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

010752 010260 000006                     MOV     R2,RMDA(R0)     ;LOAD RMDA

010756 010260 000032                     MOV     R2,RMOF(R0)     ;LOAD RMOF

010762 010260 000042                     MOV     R2,RMER2(R0)    ;LOAD RMER2

010766 010360 000016                     MOV     R3,RMAS(R0)     ;LOAD RMAS

010772 010360 000030                     MOV     R3,RMSN(R0)     ;LOAD RMSN

010776 010360 000040                     MOV     R3,RMMR2(R0)    ;LOAD RMMR2

011002 016037 000006 001334              MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER

011010 016037 000032 001360              MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER

011016 016037 000042 001370              MOV     RMER2(R0),RMER2I         ;STORE RMER2 IN INPUT BUFFER
011024 020337 001334                     CMP     R3,RMDAI
011030 001015                            BNE     20$
011032 052737 161577 001360              BIS     #XNUOF,RMOFI
011040 020337 001360                     CMP     R3,RMOFI
011044 001007                            BNE     20$
011046 052737 001567 001370              BIS     #XNUER2,RMER2I
011054 020337 001370                     CMP     R3,RMER2I
011060 001001                            BNE     20$
011062 104011                            ERROR   11              ;REG SEL 1 IS S-A-1
011064                           20$:

                                 ;TEST REG SEL 2 FOR S-A-0
011064 012760 000040 000010              MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
011072 111160 000010                     MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

011076 010260 000006                     MOV     R2,RMDA(R0)     ;LOAD RMDA

011102 010260 000032                     MOV     R2,RMOF(R0)     ;LOAD RMOF

011106 010260 000042                     MOV     R2,RMER2(R0)    ;LOAD RMER2

011112 010360 0000[illegible]            MOV     R3,RMLA(R0)     ;LOAD RMLA

011116 010360 000036                     MOV     R3,RMHR(R0)     ;LOAD RMHR
```

```

011122  010360  000046                          MOV     R3,RMEC2(R0)        ;LOAD RMEC2

011126  016037  000006  001334                  MOV     RMDA(R0),RMDAI      ;STORE RMDA IN INPUT BUFFER

011134  016037  000032  001360                  MOV     RMOF(R0),RMOFI      ;STORE RMOF IN INPUT BUFFER

011142  016037  000042  001370                  MOV     RMER2(R0),RMER2I            ;STORE RMER2 IN INPUT BUFFER
011150  020337  001334                          CMP     R3,RMDAI
011154  001015                                  BNE     30$
011156  052737  161577  001360                  BIS     #XNUOF,RMOFI
011164  020337  001360                          CMP     R3,RMOFI
011170  001007                                  BNE     30$
011172  052737  001567  001370                  BIS     #XNUER2,RMER2I
011200  020337  001370                          CMP     R3,RMER2I
011204  001001                                  BNE     30$
011206  104012                                  ERROR   12                  ;REG SEL 2 IS S-A-0
011210                                  30$:

                                        ;TEST REG SEL 2 FOR S-A-1
011210  012760  000040  000010                  MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
011216  111160  000010                          MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

011222  010260  000014                          MOV     R2,RMER1(R0)        ;LOAD RMER1

011226  010260  000034                          MOV     R2,RMDC(R0)         ;LOAD RMDC

011232  012760  000076  000000                  MOV     #ILF76,RMCS1(R0)            ;LOAD RMCS1

011240  010360  000030                          MOV     R3,RMSN(R0)         ;LOAD RMSN

011244  016037  000014  001342                  MOV     RMER1(R0),RMER1I            ;STORE RMER1 IN INPUT BUFFER

011252  016037  000034  001362                  MOV     RMDC(R0),RMDCI      ;STORE RMDC IN INPUT BUFFER
011260  052737  177701  001342                  BIS     #^CILF76,RMER1I
011266  020337  001342                          CMP     R3,RMER1I
011272  001007                                  BNE     40$
011274  052737  176000  001362                  BIS     #XNUDC,RMDCI
011302  020337  001362                          CMP     R3,RMDCI
011306  001001                                  BNE     40$
011310  104013                                  ERROR   13                  ;REG SEL 2 IS S-A-1
011312                                  40$:

                                        ;TEST REG SEL 4 FOR S-A-0
011312  012760  000040  000010                  MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
011320  111160  000010                          MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

011324  010260  000014                          MOV     R2,RMER1(R0)        ;LOAD RMER1

011330  010260  000032                          MOV     R2,RMOF(R0)         ;LOAD RMOF

011334  010260  000034                          MOV     R2,RMDC(R0)         ;LOAD RMDC

011340  010360  000026                          MOV     R3,RMDT(R0)         ;LOAD RMDT

011344  010360  000042                          MOV     R3,RMER2(R0)        ;LOAD RMER2
```

```
011350  010360  000044                  MOV     R3,RMEC1(R0)    ;LOAD RMEC1

011354  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER

011362  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER

011370  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
011376  020337  001342                  CMP     R3,RMER1I
011402  001015                          BNE     50$
011404  052737  161577  001360          BIS     #XNUOF,RMOFI
011412  020337  001360                  CMP     R3,RMOFI
011416  001007                          BNE     50$
011420  052737  176000  001362          BIS     #XNUDC,RMDCI
011426  020337  001362                  CMP     R3,RMDCI
011432  001001                          BNE     50$
011434  104014                          ERROR   14              ;REG SEL 4 IS S-A-0
011436                          50$:

                                ;TEST REG SEL 4 FOR S-A-1
011436  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
011444  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

011450  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA

011454  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2

011460  010360  000012                  MOV     R3,RMDS(R0)     ;LOAD RMDS

011464  010360  000032                  MOV     R3,RMOF(R0)     ;LOAD RMOF

011470  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER

011476  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
011504  020337  001334                  CMP     R3,RMDAI
011510  001007                          BNE     60$
011512  052737  001567  001370          BIS     #XNUER2,RMER2I
011520  020337  001370                  CMP     R3,RMER2I
011524  001001                          BNE     60$
011526  104015                          ERROR   15              ;REG SEL 4 IS S-A-1
011530                          60$:

                                ;TEST REG SEL 8 FOR S-A-0
011530  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
011536  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

011542  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1

011546  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA

011552  010360  000034                  MOV     R3,RMDC(R0)     ;LOAD RMDC

011556  010360  000042                  MOV     R3,RMER2(R0)    ;LOAD RMER2

011562  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER

```

```
011570 016037 000006 001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
011576 020337 001342                 CMP     R3,RMER1I
011602 001004                        BNE     70$
011604 020337 001334                 CMP     R3,RMDAI
011610 001001                        BNE     70$
011612 104016                        ERROR   16              ;REG SEL 8 IS S-A-0
011614                       70$:

                             ;TEST REG SEL 8 FOR S-A-1
011614 012760 000040 000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
011622 111160 000010                 MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

011626 010260 000032                 MOV     R2,RMOF(R0)     ;LOAD RMOF

011632 010260 000034                 MOV     R2,RMDC(R0)     ;LOAD RMDC

011636 010260 000042                 MOV     R2,RMER2(R0)    ;LOAD RMER2

011642 010360 000012                 MOV     R3,RMDS(R0)     ;LOAD RMDS

011646 010360 000014                 MOV     R3,RMER1(R0)    ;LOAD RMER1

011652 010360 000006                 MOV     R3,RMDA(R0)     ;LOAD RMDA

011656 016037 000032 001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER

011664 016037 000034 001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER

011672 016037 000042 001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
011700 052737 161577 001360          BIS     #XNUOF,RMOFI
011706 001015                        BNE     80$
011710 022737 176000 001362          CMP     #XNUDC,RMDCI
011716 020337 001362                 CMP     R3,RMDCI
011722 001007                        BNE     80$
011724 052737 001567 001370          BIS     #XNUER2,RMER2I
011732 020337 001370                 CMP     R3,RMER2I
011736 001001                        BNE     80$
011740 104017                        ERROR   17              ;REG SEL 8 IS S-A-1
011742                       80$:

                             ;REGISTER SELECT 16 IS TESTED BY THE ILR TEST

                             ;;*****************************************************
                             ;*TEST 7         DRIVE TYPE TEST

                             ;;*****************************************************
011742 000004                TST7:   SCOPE
011744 012737 000007 001226          MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

011752 000240                        NOP
011754 012737 000024 001120          MOV     #20.,$ICNT      ;20 ITERATIONS
011762 112737 000001 001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
011770 012737 012004 001122          MOV     #T7,$LPADR  ;LOAD LOOP ON TEST ADDRESS
011776 012737 012004 001124          MOV     #T7,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
012004                       T7:
012004 012706 001100                 MOV     #STACK,SP       ;LOAD THE STACK POINTER
```

```
012010  013700  001276                     MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
012014  013701  001456                     MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
012020  016037  000026  001142             MOV     RMDT(R0),$BDDAT     ;STORE RMDT AT $BDDAT
012026  022737  020024  001142             CMP     #SNGPRT,$BDDAT      ;SINGLE PORT RM03??
012034  001422                             BEQ     10$                 ;YES!!
012036  022737  024024  001142             CMP     #DULPRT,$BDDAT      ;DUAL PORT RM03??
012044  001416                             BEQ     10$                 ;YES!!
012046  012737  020024  001174             MOV     #SNGPRT,$TMP0       ;LOAD ACCEPTABLE VALUES
012054  012737  024024  001176             MOV     #DULPRT,$TMP1
012062  010037  001136                     MOV     R0,$BDADR           ;LOAD BAD ADDRESS
012066  062737  000026  001136             ADD     #RMDT,$BDADR
012074  104057                             ERROR   57                  ;DEVICE NOT AN RM03
012076  000137  057050                     JMP     $EOSP               ;GO TO NEXT DEVICE
012102                             10$:

                                   ;;*****************************************************************
                                   ;*TEST 10         DEVICE AVAILABLE TEST

                                   ;;*****************************************************************
012102  000004                     TST10:  SCOPE
012104  012737  000010  001226             MOV     #10,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX

012112  000240                             NOP
012114  012737  000024  001120             MOV     #20.,$ICNT          ;20 ITERATIONS
012122  112737  000001  001131             MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
012130  012737  012144  001122             MOV     #T10,$LPADR  ;LOAD LOOP ON TEST ADDRESS
012136  012737  012144  001124             MOV     #T10,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
012144                             T10:
012144  012706  001100                     MOV     #STACK,SP           ;LOAD THE STACK POINTER
012150  013700  001276                     MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
012154  013701  001456                     MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
012160  012760  000040  000010             MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
012166  111160  000010                     MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

012172  016037  000000  001142             MOV     RMCS1(R0),$BDDAT          ;STORE RMCS1 AT $BDDAT
012200  042737  173777  001142             BIC     #↑CDVA,$BDDAT       ;CLEAR ALL BUT DVA
012206  001006                             BNE     10$                 ;BRANCH IF DVA SET
012210  012737  004000  001140             MOV     #DVA,$GDDAT         ;SETUP EXPECTED
012216  010037  001136                     MOV     R0,$BDADR           ;SETUP REG ADDRESS
012222  104060                             ERROR   60                  ;DEVICE NOT AVAILABLE
012224                             10$:

                                   ;;*****************************************************************
                                   ;*TEST 11         HOLDING REGISTER TRANSFER TEST

                                   ;;*****************************************************************
012224  000004                     TST11:  SCOPE
012226  012737  000011  001226             MOV     #11,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX

012234  000240                             NOP
012236  012737  000024  001120             MOV     #20.,$ICNT          ;20 ITERATIONS
012244  112737  000001  001131             MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
012252  012737  012266  001122             MOV     #T11,$LPADR  ;LOAD LOOP ON TEST ADDRESS
012260  012737  012266  001124             MOV     #T11,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
012266                             T11:
```

```
012266  012706  001100                  MOV     #STACK,SP         ;LOAD THE STACK POINTER
012272  013700  001276                  MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
012276  013701  001456                  MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
012302  012760  000040  000010          MOV     #CLR,RMCS2(R0)    ;CLEAR THE MASSBUS
012310  111160  000010                  MOVB    (R1),RMCS2(R0)    ;SELECT UNIT
012314  005003                          CLR     R3                ;CLEAR ERROR FLAGS
012316  010037  001136                  MOV     R0,$BDADR         ;SETUP REGISTER ADDRESS
012322  062737  000036  001136          ADD     #RMHR,$BDADR
                                ;WRITE ONES THEN ZEROS IN RMHR AND CHECK FOR S-A-1 BITS.
                                ;NOTE THAT IT IS NECESSARY TO WRITE SOME OTHER REGISTER IN
                                ;ORDER TO WRITE THE HOLDING REGISTER, AND RMDA IS USED FOR THIS
                                ;PURPOSE.

012330  012760  177777  000006          MOV     #-1,RMDA(R0)      ;LOAD RMDA

012336  012760  000000  000006          MOV     #0,RMDA(R0)       ;LOAD RMDA

012344  016037  000036  001142          MOV     RMHR(R0),$BDDAT   ;STORE RMHR AT $BDDAT
012352  005137  001142                  COM     $BDDAT            ;ANY ERROR??
012356  001405                          BEQ     10$               ;NO!!
012360  005037  001140                  CLR     $GDDAT            ;LOAD EXPECTED
012364  104061                          ERROR   61
012366  052703  000001                  BIS     #BIT0,R3          ;SET ERROR FLAGS
012372                          10$:
                                ;;*********************************************************
                                ;WRITE ZEROS THEN ONES IN RMHR AND CHECK FOR S-A-0 BITS.

012372  012760  000000  000006          MOV     #0,RMDA(R0)       ;LOAD RMDA

012400  012760  177777  000006          MOV     #-1,RMDA(R0)      ;LOAD RMDA

012406  016037  000036  001142          MOV     RMHR(R0),$BDDAT   ;STORE RMHR AT $BDDAT
012414  005137  001142                  COM     $BDDAT            ;RMHR IS COMPLEMENTED WHEN READ
012420  012737  177777  001140          MOV     #-1,$GDDAT        ;SETUP EXPECTED
012426  023737  001140  001142          CMP     $GDDAT,$BDDAT     ;ANY ERROR??
012434  001403                          BEQ     20$               ;NO!!
012436  104062                          ERROR   62
012440  052703  000002                  BIS     #BIT1,R3          ;SET ERROR FLAG
012444                          20$:
                                ;;*********************************************************
                                ;IF NO PREVIOUS ERRORS, WRITE AND READ SHIFTING ONE BIT PATTERN.
012444  005703                          TST     R3                ;ANY FLAGS SET??
012446  001025                          BNE     50$               ;YES!!
012450  012702  000001                  MOV     #1,R2             ;R2=DATA PATTERN

012454                          30$:

012454  012760  000000  000006          MOV     #0,RMDA(R0)       ;LOAD RMDA

012462  010260  000006                  MOV     R2,RMDA(R0)       ;LOAD RMDA

012466  016037  000036  001142          MOV     RMHR(R0),$BDDAT   ;STORE RMHR AT $BDDAT
012474  005137  001142                  COM     $BDDAT            ;RMHR IS COMPLEMENTED
012500  023702  001142                  CMP     $BDDAT,R2         ;ANY ERROR??
012504  001404                          BEQ     40$               ;NO!!
012506  010237  001140                  MOV     R2,$GDDAT         ;SETUP EXPECTED
```

```
012512	104063				ERROR	63
012514	000402				BR	50$		;DO NOT COLLECT ALL ERRORS

012516	006302			40$:	ASL	R2		;SHIFT TO NEXT PATTERN
012520	001355				BNE	30$		;CONTINUE IF NOT DONE

012522				50$:

				;;*****************************************************************
				;*TEST 12		CONTROL STATUS #1 TRANSFER TEST

				;;*****************************************************************
012522	000004			TST12:	SCOPE
012524	012737	000012	001226		MOV	#12,$TESTN	;;SET TEST NUMBER IN APT MAIL BOX

012532	000240				NOP
012534	012737	000024	001120		MOV	#20.,$ICNT	;20 ITERATIONS
012542	112737	000001	001131		MOVB	#1,$ERMAX	;ONE ERROR ALLOWED
012550	012737	012564	001122		MOV	#T12,$LPADR	;LOAD LOOP ON TEST ADDRESS
012556	012737	012564	001124		MOV	#T12,$LPERR	;LOAD LOOP ON ERROR ADDRESS
012564				T12:
012564	012706	001100			MOV	#STACK,SP	;LOAD THE STACK POINTER
012570	013700	001276			MOV	$BASE,R0	;R0 = UNIBUS ADDRESS OF UUT
012574	013701	001456			MOV	TSTQUE,R1	;R1 = POINTER TO DEVICE
012600	005003				CLR	R3		;R3 = ERROR INDICATOR
012602	012760	000040	000010		MOV	#CLR,RMCS2(R0)	;CLEAR THE MASSBUS
012610	111160	000010			MOVB	(R1),RMCS2(R0)	;SELECT UNIT
				;WRITE ONES IN RMCS1, BITS 01-05, THEN CLEAR. READ AND
				;CHECK FOR S-A-1 BITS.

012614	012760	000076	000000		MOV	#ILF76,RMCS1(R0)	;LOAD RMCS1
012622	012760	000040	000010		MOV	#CLR,RMCS2(R0)	;CLEAR THE MASSBUS
012630	111160	000010			MOVB	(R1),RMCS2(R0)	;SELECT UNIT

012634	016037	000000	001142		MOV	RMCS1(R0),$BDDAT	;STORE RMCS1 AT $BDDAT
012642	042737	177701	001142		BIC	#↑CILF76,$BDDAT
012650	001410				BEQ	5$
012652	005037	001140			CLR	$GDDAT
012656	010037	001136			MOV	R0,$BDADR
012662	062737	000000	001136		ADD	#RMCS1,$BDADR
012670	104043				ERROR	43		;CANT CLEAR RMCS1
012672				5$:
				;WRITE ONES IN RMCS1, BITS 01-05, THEN WRITE ZEROS. READ AND CHECK FOR
				;S-A-1 BITS.

012672	012760	000076	000000		MOV	#ILF76,RMCS1(R0)	;LOAD RMCS1

012700	012760	000000	000000		MOV	#0,RMCS1(R0)	;LOAD RMCS1

012706	016037	000000	001142		MOV	RMCS1(R0),$BDDAT	;STORE RMCS1 AT $BDDAT
012714	042737	177701	001142		BIC	#↑CILF76,$BDDAT
012722	001412				BEQ	10$
012724	005037	001140			CLR	$GDDAT
012730	010037	001136			MOV	R0,$BDADR
012734	062737	000000	001136		ADD	#RMCS1,$BDADR
012742	104023				ERROR	23		;CANT WRITE 0'S
```

```
012744  052703  000001                  BIS     #BIT0,R3                ;SET ERROR FLAG
012750                          10$:
                                ;WRITE ZEROS IN RMCS1, THEN ONES, READ AND CHECK S-A-0 BITS.

012750  012760  000000  000000          MOV     #0,RMCS1(R0)            ;LOAD RMCS1

012756  012760  000076  000000          MOV     #ILF76,RMCS1(R0)                ;LOAD RMCS1

012764  016037  000000  001142          MOV     RMCS1(R0),$BDDAT                ;STORE RMCS1 AT $BDDAT
012772  042737  177701  001142          BIC     #^CILF76,$BDDAT
013000  012737  000076  001140          MOV     #ILF76,$GDDAT
013006  023737  001140  001142          CMP     $GDDAT,$BDDAT
013014  001410                          BEQ     20$
013016  010037  001136                  MOV     R0,$BDADR
013022  062737  000000  001136          ADD     #RMCS1,$BDADR
013030  104024                          ERROR   24                      ;CANT WRITE ONES
013032  052703  000002                  BIS     #BIT1,R3                ;SET ERROR FLAG
013036                          20$:
                                ;WRITE A SHIFTING ONE BIT PATTERN IN RMCS1, READ AND CHECK FOR STUCK BITS.
013036  005703                          TST     R3                      ;OMIT IF ANY ERRORS
013040  001035                          BNE     50$
013042  012702  000002                  MOV     #2,R2                   ;R2 = TEST PATTERN
013046                          30$:
013046  010203                          MOV     R2,R3                   ;R3 = EXPECTED RESULT, BITS 1-5
013050  042703  177701                  BIC     #^CILF76,R3

013054  012760  000000  000000          MOV     #0,RMCS1(R0)            ;LOAD RMCS1

013062  010260  000000                  MOV     R2,RMCS1(R0)            ;LOAD RMCS1

013066  016037  000000  001142          MOV     RMCS1(R0),$BDDAT                ;STORE RMCS1 AT $BDDAT
013074  042737  177701  001142          BIC     #^CILF76,$BDDAT
013102  020337  001142                  CMP     R3,$BDDAT
013106  001410                          BEQ     40$
013110  010337  001140                  MOV     R3,$GDDAT
013114  010037  001136                  MOV     R0,$BDADR
013120  062737  000000  001136          ADD     #RMCS1,$BDADR
013126  104025                          ERROR   25                      ;CANT WRITE SHIFTING ONES
013130  006302                  40$:    ASL     R2                      ;SHIFT TO NEXT BIT
013132  001345                          BNE     30$                     ;CONTINUE IF R2 NOT ZERO
013134                          50$:

                                ;;*****************************************************************
                                ;*TEST 13        ERROR REGISTER 1 TRANSFER TEST

                                ;;*****************************************************************
013134  000004                  TST13:  SCOPE
013136  012737  000013  001226          MOV     #13,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

013144  000240                          NOP
013146  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
013154  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
013162  012737  013176  001122          MOV     #T13,$LPADR     ;LOAD LOOP ON TEST ADDRESS
013170  012737  013176  001124          MOV     #T13,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
013176                          T13:
013176  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
```

```
013202  013700  001276                     MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
013206  013701  001456                     MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
013212  005003                             CLR     R3                  ;CLEAR ERROR FLAG
                                   ;WRITE ONES IN RMER1, CLEAR AND CHECK FOR S-A-1 BITS

013214  012760  177777  000014             MOV     #-1,RMER1(R0)       ;LOAD RMER1
013222  012760  000040  000010             MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
013230  111160  000010                     MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

013234  016037  000014  001342             MOV     RMER1(R0),RMER1I         ;STORE RMER1 IN INPUT BUFFER
013242  013737  001342  001142             MOV     RMER1I,$BDDAT
013250  042737  177760  001142             BIC     #↑C<PAR!RMR!ILF!ILR>,$BDDAT
013256  001410                             BEQ     10$
013260  005037  001140                     CLR     $GDDAT
013264  010037  001136                     MOV     R0,$BDADR
013270  062737  000014  001136             ADD     #RMER1,$BDADR
013276  104027                             ERROR   27                  ;CANT CLEAR RMER1
013300                             10$:
013300  013737  001342  001142             MOV     RMER1I,$BDDAT
013306  042737  074017  001142             BIC     #↑C<DCK!IAE!AOE!HCRC!HCE!ECH!WCF!FER>,$BDDAT
013314  001410                             BEQ     20$
013316  005037  001140                     CLR     $GDDAT
013322  010037  001136                     MOV     R0,$BDADR
013326  062737  000014  001136             ADD     #RMER1,$BDADR
013334  104030                             ERROR   30
013336                             20$:
013336  013737  001342  001142             MOV     RMER1I,$BDDAT
013344  042737  147777  001142             BIC     #↑C<OPI!DTE>,$BDDAT
013352  001410                             BEQ     30$
013354  005037  001140                     CLR     $GDDAT
013360  010037  001136                     MOV     R0,$BDADR
013364  062737  000014  001136             ADD     #RMER1,$BDADR
013372  104031                             ERROR   31
013374                             30$:
                                   ;WRITE ONES THEN ZEROS IN RMER1, READ AND CHECK FOR S-A-1 BITS

013374  012760  177777  000014             MOV     #-1,RMER1(R0)       ;LOAD RMER1

013402  012760  000000  000014             MOV     #0,RMER1(R0)        ;LOAD RMER1

013410  016037  000014  001342             MOV     RMER1(R0),RMER1I         ;STORE RMER1 IN INPUT BUFFER
013416  013737  001342  001142             MOV     RMER1I,$BDDAT
013424  042737  177770  001142             BIC     #↑C<RMR!ILF!ILR>,$BDDAT
013432  001412                             BEQ     40$
013434  005037  001140                     CLR     $GDDAT
013440  010037  001136                     MOV     R0,$BDADR
013444  062737  000014  001136             ADD     #RMER1,$BDADR
013452  104032                             ERROR   32
013454  052703  000001                     BIS     #BIT0,R3            ;SET ERROR FLAG
013460  013737  001342  001142     40$:    MOV     RMER1I,$BDDAT
013466  042737  074017  001142             BIC     #↑C<DCK!IAE!AOE!HCRC!HCE!ECH!WCF!FER>,$BDDAT
013474  001412                             BEQ     50$
013476  005037  001140                     CLR     $GDDAT
013502  010037  001136                     MOV     R0,$BDADR
013506  062737  000014  001136             ADD     #RMER1,$BDADR
013514  104033                             ERROR   33
```

```
013516  052703  000001                      BIS     #BIT0,R3            ;SET ERROR FLAG
013522                              50$:
013522  013737  001342  001142              MOV     RMER1I,$BDDAT
013530  042737  147777  001142              BIC     #↑C<OPI!DTE>,$BDDAT
013536  001412                              BEQ     60$
013540  005037  001140                      CLR     $GDDAT
013544  010037  001136                      MOV     R0,$BDADR
013550  062737  000014  001136              ADD     #RMER1,$BDADR
013556  104034                              ERROR   34
013560  052703  030000                      BIS     #BIT,R3
013564                              60$:
                                    ;WRITE ZEROS THEN ONES IN RMER1,READ AND CHECK FOR S-A-0 BITS

013564  012760  000000  000014              MOV     #0,RMER1(R0)        ;LOAD RMER1

013572  012760  177777  000014              MOV     #-1,RMER1(R0)       ;LOAD RMER1

013600  016037  000014  001142              MOV     RMER1(R0),$BDDAT            ;STORE RMER1 AT $BDDAT
013606  012737  177777  001140              MOV     #-1,$GDDAT
013614  023737  001140  001142              CMP     $GDDAT,$BDDAT
013622  001410                              BEQ     70$
013624  010037  001136                      MOV     R0,$BDADR
013630  062737  000014  001136              ADD     #RMER1,$BDADR
013636  104035                              ERROR   35
013640  052703  000002                      BIS     #BIT1,R3
013644                              70$:
                                    ;WRITE A SHIFTING 1 BIT IN RMER1 AND CHECK FOR STUCK BITS
                                    ;NOTE:  DONT TEST UNSAFE OR PARITY
013644  005703                              TST     R3                  ;SKIP THIS PART IF ANY ERRORS
013646  001045                              BNE     120$
013650  012702  000001                      MOV     #1,R2               ;R2 = TEST PATTERN
013654                              80$:
013654  012760  000040  000010              MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
013662  111160  000010                      MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

013666  010260  000014                      MOV     R2,RMER1(R0)        ;LOAD RMER1

013672  016037  000014  001142              MOV     RMER1(R0),$BDDAT            ;STORE RMER1 AT $BDDAT
013700  032702  000010                      BIT     #PAR,R2             ;DONT TEST PAR = 0
013704  001003                              BNE     90$
013706  042737  000010  001142              BIC     #PAR,$BDDAT
013714  032702  040000              90$:    BIT     #UNS,R2             ;DONT TEST UNS = 0
013720  001003                              BNE     100$
013722  042737  040000  001142              BIC     #UNS,$BDDAT
013730  020237  001142              100$:   CMP     R2,$BDDAT
013734  001410                              BEQ     110$
013736  010237  001140                      MOV     R2,$GDDAT
013742  010037  001136                      MOV     R0,$BDADR
013746  062737  000014  001136              ADD     #RMER1,$BDADR
013754  104036                              ERROR   36

013756  006302                      110$:   ASL     R2                  ;SHIFT TO NEXT BIT
013760  001335                              BNE     80$                 ;CONTINUE IF R2 NOT ZERO
013762                              120$:
```

```
                                        ;;*********************************************************************
                                        ;*TEST 14        ERROR REGISTER 2 TRANSFER TEST

                                        ;;*********************************************************************
 013762  000004                         TST14:  SCOPE
 013764  012737  000014  001226                 MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

 013772  000240                                 NOP
 013774  012737  000024  001120                 MOV     #20.,$ICNT      ;20 ITERATIONS
 014002  112737  000001  001131                 MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
 014010  012737  014024  001122                 MOV     #T14,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 014016  012737  014024  001124                 MOV     #T14,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 014024                                 T14:
 014024  012706  001100                         MOV     #STACK,SP       ;LOAD THE STACK POINTER
 014030  013700  001276                         MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
 014034  013701  001456                         MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
 014040  005003                                 CLR     R3              ;RESET ERROR FLAGS
 014042  010037  001136                         MOV     R0,$BDADR       ;SETUP BAD ADDRESS
 014046  062737  000042  001136                 ADD     #RMER2,$BDADR
 014054  005037  001140                         CLR     $GDDAT          ;SETUP EXPECTED DATA
                                        ;;*********************************************************************
                                        ;WRITE ONES IN RMER2, CLEAR AND CHECK FOR S-A-1 BITS

 014060  012760  177777  000042                 MOV     #-1,RMER2(R0)   ;LOAD RMER2
 014066  012760  000040  000010                 MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
 014074  111160  000010                         MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

 014100  016037  000042  001370                 MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER

                                        ;       TEST UNUSED BITS FOR ZERO-FAILURE ON IF
 014106  013737  001370  001142                 MOV     RMER2I,$BDDAT
 014114  042737  176210  001142                 BIC     #↑CXNUER2,$BDDAT
 014122  001403                                 BEQ     10$             ;BRANCH IF NO ERROR
 014124  104044                                 ERROR   44              ;UNUSED BITS NOT ZERO
 014126  052703  000001                         BIS     #BIT0,R3        ;SET ERROR FLAG
 014132                                 10$:
                                        ;       TEST "OPE", "IVC", "LSC" FOR ZERO-FAILURE ON CS, IF
 014132  013737  001370  001142                 MOV     RMER2I,$BDDAT
 014140  042737  143777  001142                 BIC     #↑C<OPE!IVC!LSC>,$BDDAT
 014146  001403                                 BEQ     20$             ;BRANCH IF NO ERROR
 014150  104045                                 ERROR   45              ;CANT CLEAR OPE, IVC, LSC
 014152  052703  000001                         BIS     #BIT0,R3        ;SET ERROR FLAG
 014156                                 20$:
                                        ;       TEST "LBC", "DPE" FOR ZERO-FAILURE ON DS, IF
 014156  013737  001370  001142                 MOV     RMER2I,$BDDAT
 014164  042737  175767  001142                 BIC     #↑C<LBC!DPE>,$BDDAT
 014172  001403                                 BEQ     30$             ;BRANCH IF NO ERROR
 014174  104046                                 ERROR   46              ;CANT CLEAR LBC, DPE
 014176  052703  000001                         BIS     #BIT0,R3        ;SET ERROR FLAG
 014202                                 30$:

                                        ;;*********************************************************************
                                        ;WRITE ONES IN RMER2 THEN WRITE ZEROS AND CHECK FOR S-A-1 BITS.

 014202  012760  177777  000042                 MOV     #-1,RMER2(R0)   ;LOAD RMER2

```

```
6353 014210 012760 000000 000042          MOV     #0,RMER2(R0)      ;LOAD RMER2
6354
6355 014216 016037 000042 001370          MOV     RMER2(R0),RMER2I         ;STORE RMER2 IN INPUT BUFFER
6356
6357                                   ;  TEST "OPE", "IVC", "LSC" FOR ZERO-FAILURE ON CS, IF
6358 014224 013737 001370 001142          MOV     RMER2I,$BDDAT
6359 014232 042737 143777 001142          BIC     #↑C<OPE!IVC!LSC>,$BDDAT
```

```
 6360 014240 001403                     BEQ     40$                     ;BRANCH IF NO ERROR
```

```
6361 014242 104047                      ERROR   47              ;CANT WRITE ZEROS
6362 014244 052703 000001               BIS     #BIT0,R3        ;SET ERROR FLAG
6363 014250                      40$:
6364                             ;      TEST "LBC", "DPE" FOR ZERO-FAILURE ON DS, IF
6365 014250 013737 001370 001142        MOV     RMER2I,$BDDAT
```

```
014256 042737 175767 001142           BIC     #↑C<LBC!DPE>,$BDDAT
014264 001403                         BEQ     50$                 ;BRANCH IF NO ERROR
014266 104050                         ERROR   50                  ;CANT WRITE ZEROS
014270 052703 000001                  BIS     #BIT0,R3            ;SET ERROR FLAG
014274                         50$:

                               ;;*************************************************************
                               ;WRITE ZEROS IN RMER2 THEN WRITE ONES AND CHECK FOR S-A-0 BITS.

014274 012760 000000 000042           MOV     #0,RMER2(R0)        ;LOAD RMER2

014302 012760 177777 000042           MOV     #-1,RMER2(R0)       ;LOAD RMER2

014310 016037 000042 001370           MOV     RMER2(R0),RMER2I          ;STORE RMER2 IN INPUT BUFFER
                               ;      TEST UNUSED BITS FOR ZERO-FAILURE ON IF
014316 013737 001370 001142           MOV     RMER2I,$BDDAT
014324 042737 176210 001142           BIC     #↑CXNUER2,$BDDAT
014332 001403                         BEQ     60$                 ;BRANCH IF NO ERROR
014334 104044                         ERROR   44                  ;UNUSED BITS NOT ZERO
014336 052703 000001                  BIS     #BIT0,R3            ;SET ERROR FLAG
014342                         60$:
                               ;      TEST USED BITS FOR ONE-FAILURE ON IF
014342 013737 001370 001142           MOV     RMER2I,$BDDAT
014350 042737 001567 001142           BIC     #XNUER2,$BDDAT
014356 012737 176210 001140           MOV     #↑CXNUER2,$GDDAT
014364 023737 001140 001142           CMP     $GDDAT,$BDDAT
014372 001403                         BEQ     70$                 ;BRANCH IF NO ERROR
014374 104051                         ERROR   51                  ;CANT WRITE ONES
014376 052703 000002                  BIS     #BIT1,R3            ;SET ERROR FLAG
014402                         70$:

                               ;;*************************************************************
                               ;IF NO PREVIOUS ERROR, TEST BIT INTERFERENCE WITH SHIFTING ONE BIT.
014402 005703                         TST     R3                  ;ANY ERRORS?
014404 001044                         BNE     120$                ;YES!!
014406 012702 000001                  MOV     #1,R2               ;R2=DATA PATTERN
014412                         80$:
014412 012760 000040 000010           MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
014420 111160 000010                  MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

014424 010260 000042                  MOV     R2,RMER2(R0)        ;LOAD RMER2

014430 016037 000042 001142           MOV     RMER2(R0),$BDDAT          ;STORE RMER2 AT $BDDAT
014436 032702 040000                  BIT     #SKI,R2             ;IS SKI BEING SET?
014442 001003                         BNE     90$                 ;YES!!
014444 042737 040000 001142           BIC     #SKI,$BDDAT         ;DONT TEST SKI FOR ZERO
014452 032702 000200           90$:   BIT     #DVC,R2             ;IS DVC BEING SET??
014456 001003                         BNE     100$                ;YES!!
014460 042737 000200 001142           BIC     #DVC,$BDDAT         ;DONT TEST DVC FOR ZERO
014466 010237 001140           100$:  MOV     R2,$GDDAT
014472 042737 001567 001140           BIC     #XNUER2,$GDDAT      ;UNUSED BITS SHOULD BE ZERO
014500 023737 001140 001142           CMP     $GDDAT,$BDDAT       ;ANY ERRORS??
014506 001401                         BEQ     110$                ;NO!!
014510 104052                         ERROR   52                  ;CANT WRITE SHIFTING ONES

014512 006302                  110$:  ASL     R2                  ;SHIFT TO NEXT DATA BIT
```

```
014514  001336                          BNE     80$             ;CONTINUE IF NOT DONE
014516                          120$:
                                ;;*****************************************************
                                ;*TEST 15        CLEAR OFFSET STUCK ACTIVE TEST
                                ;;*****************************************************
014516  000004                  TST15:  SCOPE
014520  012737  000015  001226          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
014526  000240                          NOP
014530  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
014536  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
014544  012737  014560  001122          MOV     #T15,$LPADR  ;LOAD LOOP ON TEST ADDRESS
014552  012737  014560  001124          MOV     #T15,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
014560                          T15:
014560  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
014564  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
014570  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
014574  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
014602  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
014606  012760  177777  000032          MOV     #-1,RMOF(R0)    ;LOAD RMOF
014614  016037  000032  001142          MOV     RMOF(R0),$BDDAT ;STORE RMOF AT $BDDAT
014622  042737  177577  001142          BIC     #↑COFD,$BDDAT
014630  001011                          BNE     10$             ;BRANCH IF OFD IS A ONE
014632  012737  000200  001140          MOV     #OFD,$GDDAT     ;SETUP ERROR MESSAGE
014640  010037  001136                  MOV     R0,$BDADR
014644  062737  000032  001136          ADD     #RMOF,$BDADR
014652  104172                          ERROR   172             ;CANT SET OFD BIT
014654                          10$:                            ;END OF TEST
                                ;;*****************************************************
                                ;*TEST 16        OFFSET REGISTER TRANSFER TEST
                                ;;*****************************************************
014654  000004                  TST16:  SCOPE
014656  012737  000016  001226          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
014664  000240                          NOP
014666  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
014674  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
014702  012737  014716  001122          MOV     #T16,$LPADR  ;LOAD LOOP ON TEST ADDRESS
014710  012737  014716  001124          MOV     #T16,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
014716                          T16:
014716  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
014722  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
014726  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
014732  005003                          CLR     R3              ;RESET ERROR FLAGS
014734  010037  001136                  MOV     R0,$BDADR       ;SETUP BAD ADDRESS
014740  062737  000032  001136          ADD     #RMOF,$BDADR
014746  005037  001140                  CLR     $GDDAT          ;SETUP EXPECTED DATA
                                ;;*****************************************************
```

```
                                        ;WRITE ONES THEN ZEROS IN RMOF AND CHECK FOR S-A-1 BITS.

014752  012760  177777  000032                  MOV     #-1,RMOF(R0)    ;LOAD RMOF

014760  012760  000000  000032                  MOV     #0,RMOF(R0)     ;LOAD RMOF

014766  016037  000032  001360                  MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
                                        ;       CHECK UNUSED BITS OF RMOF FOR ZERO
014774  013737  001360  001142                  MOV     RMOFI,$BDDAT    ;GET UNUSED BITS
015002  042737  016200  001142                  BIC     #^CXNUOF,$BDDAT
015010  001403                                  BEQ     10$             ;BRANCH IF NO ERROR
015012  104053                                  ERROR   53              ;UNUSED BITS NOT ZERO RMOF
015014  052703  000001                          BIS     #BIT0,R3        ;SET ERROR FLAG
015020                                  10$:
                                        ;       CHECK USED BITS OF RMOF FOR ZERO
015020  013737  001360  001142                  MOV     RMOFI,$BDDAT    ;GET USED BITS
015026  042737  161577  001142                  BIC     #XNUOF,$BDDAT
015034  001403                                  BEQ     20$             ;BRANCH IF NO ERROR
015036  104054                                  ERROR   54              ;USED BITS RMOF S-A-1
015040  052703  000001                          BIS     #BIT0,R3        ;SET ERROR FLAG
015044                                  20$:

                                        ;;**************************************************************
                                        ;WRITE ZEROS THEN ONES ON RMOF AND CHECK FOR S-A-0 BITS.

015044  012760  000000  000032                  MOV     #0,RMOF(R0)     ;LOAD RMOF

015052  012760  177777  000032                  MOV     #-1,RMOF(R0)    ;LOAD RMOF

015060  016037  000032  001360                  MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
                                        ;       CHECK UNUSED BITS OF RMOF FOR ZERO.
015066  013737  001360  001142                  MOV     RMOFI,$BDDAT    ;GET UNUSED BITS
015074  042737  016200  001142                  BIC     #^CXNUOF,$BDDAT
015102  001403                                  BEQ     30$             ;BRANCH IF NO ERROR
015104  104053                                  ERROR   53              ;UNUSED BITS OF RMOF NOT ZERO
015106  052703  000001                          BIS     #BIT0,R3
015112                                  30$:
                                        ;       CHECK USED BITS OF RMOF FOR ONE.
015112  013737  001360  001142                  MOV     RMOFI,$BDDAT    ;GET USED BITS
015120  042737  161577  001142                  BIC     #XNUOF,$BDDAT
015126  012737  016200  001140                  MOV     #FMT16!ECI!HCI!OFD,$GDDAT
015134  023737  001140  001142                  CMP     $GDDAT,$BDDAT
015142  001403                                  BEQ     40$             ;BRANCH IF NO ERROR
015144  104055                                  ERROR   55              ;USED BITS RMOF S-A-0
015146  052703  000002                          BIS     #BIT1,R3
015152                                  40$:

                                        ;;**************************************************************
                                        ;IF NO PREVIOUS ERRORS, TEST RMOF WITH SHIFTING ONE BIT PATTERN.
015152  005703                                  TST     R3              ;ANY ERROR??
015154  001025                                  BNE     70$             ;YES!!
015156  012702  000001                          MOV     #1,R2           ;STARTING DATA PATTERN
015162                                  50$:

015162  012760  000000  000032                  MOV     #0,RMOF(R0)     ;LOAD RMOF
```

```
015170  010260  000032                          MOV     R2,RMOF(R0)             ;LOAD RMOF

015174  016037  000032  001142                  MOV     RMOF(R0),$BDDAT         ;STORE RMOF AT $BDDAT
015202  010203                                  MOV     R2,R3                   ;SETUP EXPECTED RESULT
015204  042703  161577                          BIC     #XNUOF,R3               ;CLEAR UNUSED BITS
015210  020337  001142                          CMP     R3,$BDDAT               ;COMPARE EXPECTED & RECEIVED
015214  001403                                  BEQ     60$                     ;BRANCH IF NO ERROR
015216  010337  001140                          MOV     R3,$GDDAT               ;LOAD EXPECTED
015222  104056                                  ERROR   56                      ;CANT WRITE SHIFTING 1 BIT
015224  006302                          60$:    ASL     R2                      ;SHIFT TO NEXT BIT
015226  001355                                  BNE     50$                     ;CONTINUE IF NOT DONE

015230                                  70$:

                                        ;;*****************************************************************
                                        ;*TEST 17        SERIAL NUMBER TEST

                                        ;;*****************************************************************
015230  000004                          TST17:  SCOPE
015232  012737  000017  001226                  MOV     #17,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

015240  000240                                  NOP
015242  012737  000024  001120                  MOV     #20.,$ICNT              ;20 ITERATIONS
015250  112737  000001  001131                  MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
015256  012737  015272  001122                  MOV     #T17,$LPADR     ;LOAD LOOP ON TEST ADDRESS
015264  012737  015272  001124                  MOV     #T17,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
015272                                  T17:
015272  012706  001100                          MOV     #STACK,SP               ;LOAD THE STACK POINTER
015276  013700  001276                          MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
015302  013701  001456                          MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
015306  012760  000040  000010                  MOV     #CLR,RMCS2(R0)          ;CLEAR THE MASSBUS
015314  111160  000010                          MOVB    (R1),RMCS2(R0)          ;SELECT UNIT
015320  010037  001136                          MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS FOR TYPEOUT
015324  062737  000030  001136                  ADD     #RMSN,$BDADR
015332  012702  000031                          MOV     #25.,R2                 ;READ RMSN 25 TIMES
                                        ;READ RMSN AND USE THE RESULT AS EXPECTED VALUE

015336  016037  000030  001140                  MOV     RMSN(R0),$GDDAT         ;STORE RMSN AT $GDDAT
015344                                  10$:
                                        ;READ RMSN AND COMPARE WITH INITIAL VALUE

015344  016037  000030  001142                  MOV     RMSN(R0),$BDDAT         ;STORE RMSN AT $BDDAT
015352  023737  001140  001142                  CMP     $GDDAT,$BDDAT
015360  001401                                  BEQ     20$                     ;BRANCH IF SERIAL NUMBER CONSISTENT
015362  104136                                  ERROR   136                     ;SERIAL NUMBER INCONSISTENT
015364                                  20$:
                                        ;DECREMENT COUNT AND CONTINUE IF NOT DONE
015364  005302                                  DEC     R2
015366  100366                                  BPL     10$
015370                                  30$:                                    ;END OF TEST

                                        ;;*****************************************************************
                                        ;*TEST 20        CONTROL BUS PARITY DETECTION TEST

                                        ;;*****************************************************************
015370  000004                          TST20:  SCOPE
```

```
015372  012737  000020  001226          MOV     #20,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

015400  000240                          NOP
015402  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
015410  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
015416  012737  015432  001122          MOV     #T20,$LPADR     ;LOAD LOOP ON TEST ADDRESS
015424  012737  015432  001124          MOV     #T20,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
015432                          T20:
015432  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
015436  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
015442  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE

                                ;SETUP FOR FIRST TEST LOOP (NO ERROR)
015446  005037  001140                  CLR     $GDDAT                  ;"PAR" SHOULD BE ZERO
015452  111137  001412                  MOVB    (R1),RMCS20             ;SETUP RMCS2 VALUE
015456  042737  177770  001412          BIC     #↑CUNTMSK,RMCS20
015464  012737  000001  001440          MOV     #1,RMHR0                ;INITIALIZE DATA PATTERN
015472  000402                          BR      6$                      ;SKIP INCREMENT FIRST TIME
015474  006337  001440          5$:     ASL     RMHR0                   ;SHIFT TO NEXT PATTERN
015500                          6$:
                                ;CLEAR AND VERIFY THAT "PAR" IS RESET
015500  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
015506  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

015512  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT

015520  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
015526  042737  177767  001142          BIC     #↑CPAR,$BDDAT   ;DID "PAR" RESET?
015534  001415                          BEQ     20$                     ;YES!!
015536  010037  001136                  MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS
015542  062737  000014  001136          ADD     #RMER1,$BDADR
015550  032737  000010  001370          BIT     #DPE,RMER2I             ;IS "DPE" SET??
015556  001002                          BNE     10$                     ;YES!!
015560  104067                          ERROR   67                      ;CANT CLEAR "PAR"
015562  000453                          BR      50$
015564  104070                  10$:    ERROR   70                      ;CANT CLEAR "PAR", "DPE"
015566  000451                          BR      50$
015570                          20$:

                                ;WRITE TEST PATTERN AND VERIFY "PAR" STATUS

015570  013760  001412  000010          MOV     RMCS20,RMCS2(R0)        ;LOAD RMCS2

015576  013760  001440  000036          MOV     RMHR0,RMHR(R0)  ;LOAD RMHR

015604  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
015612  042737  177767  001142          BIC     #↑CPAR,$BDDAT
015620  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS "PAR" CORRECT??
015626  001410                          BEQ     40$                     ;YES!!
015630  032737  000020  001412          BIT     #PAT,RMCS20             ;SHOULD "PAR" BE SET?
015636  001002                          BNE     30$                     ;YES!!
015640  104071                          ERROR   71                      ;"PAR" SHOULD NOT BE SET
015642  000423                          BR      50$
015644  104072                  30$:    ERROR   72                      ;"PAR" SHOULD BE SET
015646  000421                          BR      50$                     ;SKIP TO NEXT

                                ;GO TO NEXT PATTERN
015650  005737  001440          40$:    TST     RMHR0                   ;IS DATA PATTERN COMPLETE??
```

```
 015654  001307                                BNE     5$                  ;NO!!
 015656  032737  000020  001412                BIT     #PAT,RMCS20         ;IS TEST COMPLETE??
 015664  001012                                BNE     50$                 ;YES!!
                                       ;SETUP FOR SECOND TEST LOOP (ERROR)
 015666  052737  000020  001412                BIS     #PAT,RMCS20         ;TURN ON BAD PARITY
 015674  012737  000010  001140                MOV     #PAR,$GDDAT         ;EXPECT ERROR
 015702  012737  000001  001440                MOV     #1,RMHRO            ;INITIALIZE DATA PATTERN
 015710  000673                                BR      6$                  ;START LOOP

 015712                                50$:                                ;END OF TEST

                                       ;;*****************************************************************
                                       ;*TEST 21        CONTROL BUS PARITY GENERATION TEST

                                       ;;*****************************************************************
 015712  000004                        TST21:  SCOPE
 015714  012737  000021  001226                MOV     #21,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX

 015722  000240                                NOP
 015724  012737  000024  001120                MOV     #20.,$ICNT          ;20 ITERATIONS
 015732  112737  000001  001131                MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
 015740  012737  015754  001122                MOV     #T21,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 015746  012737  015754  001124                MOV     #T21,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 015754                                T21:
 015754  012706  001100                        MOV     #STACK,SP           ;LOAD THE STACK POINTER
 015760  013700  001276                        MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
 015764  013701  001456                        MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE

                                       ;SETUP FOR TEST (NO ERROR)
 015770  012737  000001  001440                MOV     #1,RMHRO            ;INITIALIZE DATA PATTERN
 015776  005037  001140                        CLR     $GDDAT              ;MCPE SHOULD BE ZERO
 016002  000402                                BR      20$                 ;DONT SHIFT FIRST TIME
 016004                                10$:
                                       ;SHIFT DATA PATTERN
 016004  006337  001440                        ASL     RMHRO
 016010                                20$:
                                       ;CLEAR, THEN TRANSFER DATA TO RM03
 016010  012760  000040  000010                MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
 016016  111160  000010                        MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

 016022  013760  001440  000036                MOV     RMHRO,RMHR(R0)  ;LOAD RMHR
                                       ;TRANSFER DATA TO RH, VERIFY NO "MCPE" ERROR

 016030  016037  000036  001364                MOV     RMHR(R0),RMHRI  ;STORE RMHR IN INPUT BUFFER

 016036  016037  000000  001142                MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
 016044  042737  157777  001142                BIC     #↑CMCPE,$BDDAT  ;WAS BAD PARITY DETECTED??
 016052  001402                                BEQ     30$                 ;NO!!
 016054  104073                                ERROR   73                  ;BAD PARITY DETECTED BY RH
 016056  000403                                BR      40$

 016060                                30$:
                                       ;GO TO NEXT PATTERN
 016060  005737  001440                        TST     RMHRO               ;DONE ALL PATTERNS??
 016064  001347                                BNE     10$                 ;NO!!
```

```
016066                              40$:                                    ;END OF TEST

                                    ;;*****************************************************************
                                    ;*TEST 22        RMDA,RMDC FAULT TEST

                                    ;;*****************************************************************
016066  000004                      TST22:  SCOPE
016070  012737  000022  001226              MOV     #22,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX

016076  000240                              NOP
016100  012737  000024  001120              MOV     #20.,$ICNT          ;20 ITERATIONS
016106  112737  000001  001131              MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
016114  012737  016130  001122              MOV     #T22,$LPADR   ;LOAD LOOP ON TEST ADDRESS
016122  012737  016130  001124              MOV     #T22,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
016130                              T22:
016130  012706  001100                      MOV     #STACK,SP           ;LOAD THE STACK POINTER
016134  013700  001276                      MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
016140  013701  001456                      MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
016144  012760  000040  000010              MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
016152  111160  000010                      MOVB    (R1),RMCS2(R0)      ;SELECT UNIT

                                    ;WRITE ZEROS, THEN ONES IN RMDA,RMDC-READ AND TEST FOR S-A-0

016156  012760  000000  000006              MOV     #0,RMDA(R0)         ;LOAD RMDA

016164  012760  000000  000034              MOV     #0,RMDC(R0)         ;LOAD RMDC

016172  012760  177777  000006              MOV     #-1,RMDA(R0)        ;LOAD RMDA

016200  012760  177777  000034              MOV     #-1,RMDC(R0)        ;LOAD RMDC

016206  016037  000006  001334              MOV     RMDA(R0),RMDAI      ;STORE RMDA IN INPUT BUFFER

016214  016037  000034  001362              MOV     RMDC(R0),RMDCI      ;STORE RMDC IN INPUT BUFFER
016222  022737  177777  001334              CMP     #-1,RMDAI           ;IS ANY REGISTER ALL ONES??
016230  001410                              BEQ     10$                 ;YES!!
016232  052737  176000  001362              BIS     #XNUDC,RMDCI        ;SET UNUSED BITS
016240  022737  177777  001362              CMP     #-1,RMDCI
016246  001401                              BEQ     10$                 ;YES!!
016250  104042                              ERROR   42                  ;RMDC AND RMDA S-A-0

016252                              10$:



                                    ;;*****************************************************************
                                    ;*TEST 23        DISK ADDRESS TRANSFER TEST

                                    ;;*****************************************************************
016252  000004                      TST23:  SCOPE
016254  012737  000023  001226              MOV     #23,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX
016262  000240                              NOP
016264  012737  000024  001120              MOV     #20.,$ICNT          ;20 ITERATIONS
016272  112737  000001  001131              MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
016300  012737  016314  001122              MOV     #T23,$LPADR   ;LOAD LOOP ON TEST ADDRESS
016306  012737  016314  001124              MOV     #T23,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
```

```
016314                             T23:
016314  012706  001100                     MOV     #STACK,SP          ;LOAD THE STACK POINTER
016320  013700  001276                     MOV     $BASE,R0           ;R0 = UNIBUS ADDRESS OF UUT
016324  013701  001456                     MOV     TSTQUE,R1          ;R1 = POINTER TO DEVICE
016330  005003                             CLR     R3                 ;R3 = ERROR INDICATOR
016332  012760  000040  000010             MOV     #CLR,RMCS2(R0)     ;CLEAR THE MASSBUS
016340  111160  000010                     MOVB    (R1),RMCS2(R0)     ;SELECT UNIT
                                   ;WRITE ONES IN RMDA, THEN WRITE ZEROS, READ BACK AND CHECK FOR S-A-1 BITS

016344  012760  177777  000006             MOV     #-1,RMDA(R0)       ;LOAD RMDA

016352  012760  000000  000006             MOV     #0,RMDA(R0)        ;LOAD RMDA

016360  016037  000006  001142             MOV     RMDA(R0),$BDDAT    ;STORE RMDA AT $BDDAT
016366  005737  001142                     TST     $BDDAT
016372  001412                             BEQ     10$
016374  005037  001140                     CLR     $GDDAT
016400  010037  001136                     MOV     R0,$BDADR
016404  062737  000006  001136             ADD     #RMDA,$BDADR
016412  104020                             ERROR   20
016414  052703  000001                     BIS     #BIT0,R3           ;SET ERROR FLAG
                                   ;WRITE ZEROS IN RMDA, THEN WRITE ONES, READ BACK AND CHECK FOR S-A-0 BITS
016420                             10$:

016420  012760  000000  000006             MOV     #0,RMDA(R0)        ;LOAD RMDA

016426  012760  177777  000006             MOV     #-1,RMDA(R0)       ;LOAD RMDA

016434  016037  000006  001142             MOV     RMDA(R0),$BDDAT    ;STORE RMDA AT $BDDAT
016442  023727  001142  177777             CMP     $BDDAT,#-1
016450  001413                             BEQ     20$
016452  012737  177777  001140             MOV     #-1,$GDDAT
016460  010037  001136                     MOV     R0,$BDADR
016464  062737  000006  001136             ADD     #RMDA,$BDADR
016472  104021                             ERROR   21
016474  052703  000002                     BIS     #BIT1,R3           ;SET ERROR FLAG
                                   ;WRITE A SHIFTING 1 BIT PATTERN IN RMDA, READ, AND CHECK FOR STUCK BITS
016500  005703                     20$:    TST     R3                 ;OMIT BIT TEST IF ANY ERROR
016502  001027                             BNE     50$
016504  012702  000001                     MOV     #1,R2              ;R2 = TEST PATTERN
016510                             30$:

016510  012760  000000  000006             MOV     #0,RMDA(R0)        ;LOAD RMDA

016516  010260  000006                     MOV     R2,RMDA(R0)        ;LOAD RMDA

016522  016037  000006  001142             MOV     RMDA(R0),$BDDAT    ;STORE RMDA AT $BDDAT
016530  023702  001142                     CMP     $BDDAT,R2
016534  001410                             BEQ     40$
016536  010237  001140                     MOV     R2,$GDDAT
016542  010037  001136                     MOV     R0,$BDADR
016546  062737  000006  001136             ADD     #RMDA,$BDADR
016554  104022                             ERROR   22
016556  006302                     40$:    ASL     R2                 ;SHIFT TO NEXT BIT
016560  001353                             BNE     30$                ;CONTINUE IF R2 NOT ZERO
016562                             50$:
```

```
                                          ;;*************************************************************
                                          ;*TEST 24        DESIRED CYLINDER TRANSFER TEST

                                          ;;*************************************************************
016562  000004                            TST24:  SCOPE
016564  012737  000024  001226                    MOV     #24,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX

016572  000240                                    NOP
016574  012737  000024  001120                    MOV     #20.,$ICNT          ;20 ITERATIONS
016602  112737  000001  001131                    MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
016610  012737  016624  001122                    MOV     #T24,$LPADR   ;LOAD LOOP ON TEST ADDRESS
016616  012737  016624  001124                    MOV     #T24,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
016624                                    T24:
016624  012706  001100                            MOV     #STACK,SP           ;LOAD THE STACK POINTER
016630  013700  001276                            MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
016634  013701  001456                            MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
016640  005003                                    CLR     R3                  ;RESET ERROR FLAGS
016642  012760  000040  000010                    MOV     #CLR,RMCS2(R0)      ;CLEAR THE MASSBUS
016650  111160  000010                            MOVB    (R1),RMCS2(R0)      ;SELECT UNIT
016654  005037  001140                            CLR     $GDDAT              ;LOAD EXPECTED
016660  010037  001136                            MOV     R0,$BDADR           ;LOAD REG ADDRESS
016664  062737  000034  001136                    ADD     #RMDC,$BDADR

                                          ;WRITE ONES IN RMDC AND VERIFY THAT UNUSED BITS ARE ZERO

016672  012760  177777  000034                    MOV     #-1,RMDC(R0)        ;LOAD RMDC

016700  016037  000034  001142                    MOV     RMDC(R0),$BDDAT ;STORE RMDC AT $BDDAT
016706  042737  001777  001142                    BIC     #^CXNUDC,$BDDAT             ;CLEAR ALL USED BITS
016714  001403                                    BEQ     5$                          ;BRANCH IF NO ERROR
016716  104134                                    ERROR   134                 ;UNUSED BITS OF RMDC NOT ZERO
016720  052703  000001                            BIS     #BIT0,R3            ;SET ERROR FLAG
016724                                    5$:
                                          ;WRITE ONES IN RMDC, THEN WRITE ZEROS, READ AND CHECK FOR S-A-1 BITS

016724  012760  177777  000034                    MOV     #-1,RMDC(R0)        ;LOAD RMDC

016732  012760  000000  000034                    MOV     #0,RMDC(R0)         ;LOAD RMDC

016740  016037  000034  001142                    MOV     RMDC(R0),$BDDAT ;STORE RMDC AT $BDDAT
016746  042737  176000  001142                    BIC     #XNUDC,$BDDAT       ;CLEAR UNUSED BITS
016754  001403                                    BEQ     10$                 ;BRANCH IF NO ERROR
016756  104037                                    ERROR   37                  ;CANT WRITE ONES RMDC
016760  052703  000001                            BIS     #BIT0,R3            ;SET ERROR FLAG
016764                                    10$:
                                          ;WRITE ZEROS, THEN ONES IN RMDC, READ AND CHECK FOR S-A-0 BITS

016764  012760  000000  000034                    MOV     #0,RMDC(R0)         ;LOAD RMDC

016772  012760  177777  000034                    MOV     #-1,RMDC(R0)        ;LOAD RMDC

017000  016037  000034  001142                    MOV     RMDC(R0),$BDDAT ;STORE RMDC AT $BDDAT
017006  052737  176000  001142                    BIS     #XNUDC,$BDDAT       ;SET UNUSED BITS
017014  012737  177777  001140                    MOV     #-1,$GDDAT          ;LOAD EXPECTED RESULT
017022  023737  001140  001142                    CMP     $GDDAT,$BDDAT       ;IS RMDC ALL ONES ??
```

```
017030  001403                          BEQ     20$                     ;YES !!
017032  104040                          ERROR   40
017034  052703  000002                  BIS     #BIT1,R3                ;SET ERROR FLAG
017040                          20$:
                                ;OMIT BIT TEST IF ANY ERRORS
017040  005703                          TST     R3
017042  001026                          BNE     60$
017044  012702  000001                  MOV     #1,R2                   ;R2 = TEST PATTERN
017050                          30$:
                                ;TEST RMDC WITH SHIFTING ONE BIT

017050  012760  000000  000034          MOV     #0,RMDC(R0)             ;LOAD RMDC

017056  010260  000034                  MOV     R2,RMDC(R0)             ;LOAD RMDC
017062  010203                          MOV     R2,R3                   ;R3 = EXPECTED RESULT
017064  042703  176000                  BIC     #XNUDC,R3               ;CLEAR ANY UNUSED BITS

017070  016037  000034  001142          MOV     RMDC(R0),$BDDAT ;STORE RMDC AT $BDDAT
017076  023703  001142                  CMP     $BDDAT,R3
017102  001404                          BEQ     50$
017104  010337  001140                  MOV     R3,$GDDAT
017110  104041                          ERROR   41
017112  000402                          BR      60$                     ;SKIP TO NEXT

017114  006302                  50$:    ASL     R2                      ;SHIFT TO NEXT BIT
017116  001354                          BNE     30$                     ;CONTINUE IF R2 NOT ZERO

017120                          60$:


                                ;;*****************************************************************
                                ;*TEST 25         ILLEGAL REGISTER TEST

                                ;;*****************************************************************
017120  000004                  TST25:  SCOPE
017122  012737  000025  001226          MOV     #25,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX
017130  000240                          NOP
017132  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
017140  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
017146  012737  017162  001122          MOV     #T25,$LPADR     ;LOAD LOOP ON TEST ADDRESS
017154  012737  017162  001124          MOV     #T25,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
017162                          T25:
017162  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
017166  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
017172  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
017176  005037  001140                  CLR     $GDDAT                  ;"ILR" SHOULD BE ZERO
017202  012702  000000                  MOV     #0,R2                   ;R2=REGISTER INDEX
017206                          10$:
                                ;;*****************************************************************
                                ;CLEAR AND VERIFY THAT "ILR" STATUS IS ZERO
017206  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
017214  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

017220  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
017226  042737  177775  001142          BIC     #↑CILR,$BDDAT
017234  001411                          BEQ     20$                     ;BRANCH IF ILR IS RESET
017236  005037  001140                  CLR     $GDDAT                  ;SETUP GOOD DATA, REG ADR
```

```
017242  010037  001136                  MOV     R0,$BDADR
017246  062737  000014  001136          ADD     #RMER1,$BDADR
017254  104064                          ERROR   64              ;CANT CLEAR ILR
017256  000551                          BR      140$
017260                          20$:
                                ;;*************************************************************
                                ;WRITE THE REGISTER (INDEX=R2) AND TEST ILR STATUS
017260  010003                          MOV     R0,R3           ;R3=REG ADDRESS
017262  060203                          ADD     R2,R3
017264  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
017270  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
017274  012737  017336  000004          MOV     #40$,ERRVEC     ;SETUP FOR BUS TIMEOUT
017302  012737  000300  000006          MOV     #PR6,ERRVEC+2
017310  005004                          CLR     R4              ;R4=REGISTER VALUE
017312  022702  000010                  CMP     #RMCS2,R2
017316  001001                          BNE     30$
017320  111104                          MOVB    (R1),R4         ;SELECT DRIVE IF RMCS2
017322  010413                  30$:    MOV     R4,(R3)         ;WRITE TEST REGISTER
017324  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
017330  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
017334  000416                          BR      60$
017336  012716  017344          40$:    MOV     #45$,(SP)       ;DUMMY RTI ADDRESS
017342  000002                          RTI                     ;RESTORE PRIORITY
017344                          45$:
017344  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
017350  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
017354  020227  000046                  CMP     R2,#RMEC2       ;WERE ALL RM03 REGISTERS READ??
017360  101003                          BHI     50$             ;YES!!
017362  010337  001136                  MOV     R3,$BDADR
017366  104074                          ERROR   74              ;UNEXPECTED BUS TIMEOUT
017370  000504                  50$:    BR      140$

017372                          60$:

017372  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
017400  042737  177775  001142          BIC     #^CILR,$BDDAT
017406  023737  001140  001142          CMP     $GDDAT,$BDDAT   ;IS "ILR" OK??
017414  001411                          BEQ     80$             ;YES!!
017416  010337  001174                  MOV     R3,$TMP0        ;SAVE ADDRESS
017422  032737  000002  001140          BIT     #ILR,$GDDAT     ;SHOULD "ILR" BE SET??
017430  001002                          BNE     70$             ;YES!!
017432  104065                          ERROR   65              ;"ILR" SHOULD NOT BE SET
017434  000401                          BR      80$
017436  104066                  70$:    ERROR   66              ;"ILR" SHOULD BE SET

017440                          80$:
                                ;ADVANCE TO THE NEXT REGISTER ADDRESS
017440  062702  000002                  ADD     #2,R2           ;INCREMENT INDEX
017444  022702  000050                  CMP     #RMBAE,R2       ;IS THIS RMBAE??
017450  001033                          BNE     100$            ;NO!!
017452  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
017456  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
017462  012737  017564  000004          MOV     #130$,ERRVEC    ;SETUP FOR TIMEOUT
017470  012737  000300  000006          MOV     #PR6,ERRVEC+2

017476  012760  001400  000000          MOV     #A17!A16,RMCS1(R0)      ;LOAD RMCS1
```

```

017504  016037  000050  001376          MOV     RMBAE(R0),RMBAEI                ;STORE RMBAE IN REGISTER INPUT BUFFER
017512  042737  177774  001376          BIC     #^C<BIT1!BIT0>,RMBAEI
017520  001003                          BNE     90$                     ;BRANCH IF RH70 CONTROLLER
017522  052737  000002  001140          BIS     #ILR,$GDDAT             ;"ILR" SHOULD BE SET
017530                          90$:
017530  012637  000006                  MOV     (SP)+,ERRVEC+2                  ;;POP STACK INTO ERRVEC+2
017534  012637  000004                  MOV     (SP)+,ERRVEC            ;;POP STACK INTO ERRVEC
017540  022702  000054          100$:   CMP     #RMCS3+2,R2             ;DONE ALL LEGAL REGISTERS
017544  001003                          BNE     110$
017546  052737  000002  001140          BIS     #ILR,$GDDAT             ;"ILR" SHOULD BE SET
017554  022702  000076          110$:   CMP     #76,R2                  ;DONE ALL TESTS
017560  103410                          BLO     140$                    ;YES!!
017562  000611                  120$:   BR      10$

017564  012716  017572          130$:   MOV     #135$,(SP)              ;DUMMY RTI ADDRESS
017570  000002                          RTI                             ;RESTORE PRIORITY
017572                          135$:
017572  012637  000004                  MOV     (SP)+,ERRVEC            ;;POP STACK INTO ERRVEC
017576  012637  000006                  MOV     (SP)+,ERRVEC+2                  ;;POP STACK INTO ERRVEC+2
017602                          140$:                                   ;END OF TEST


                                ;;*****************************************************************
                                ;*TEST 26         RESET GO BY INIT TEST

                                ;;*****************************************************************
017602  000004                  TST26:  SCOPE
017604  012737  000026  001226          MOV     #26,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

017612  000240                          NOP
017614  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
017622  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
017630  012737  017644  001122          MOV     #T26,$LPADR     ;LOAD LOOP ON TEST ADDRESS
017636  012737  017644  001124          MOV     #T26,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
017644                          T26:
017644  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
017650  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
017654  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
017660  012760  000040  000010          MOV     #CLR,RMCS2(R0)          ;CLEAR THE MASSBUS
017666  111160  000010                  MOVB    (R1),RMCS2(R0)          ;SELECT UNIT
017672  010037  001136                  MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS FOR MSG
                                ;SET GO, INITIALIZE AND VERIFY THAT GO IS RESET

017676  012760  000001  000000          MOV     #GO,RMCS1(R0)           ;LOAD RMCS1
017704  012760  000040  000010          MOV     #CLR,RMCS2(R0)          ;CLEAR THE MASSBUS
017712  111160  000010                  MOVB    (R1),RMCS2(R0)          ;SELECT UNIT

017716  016037  000000  001142          MOV     RMCS1(R0),$BDDAT                ;STORE RMCS1 AT $BDDAT
017724  042737  177776  001142          BIC     #^CGO,$BDDAT
017732  001403                          BEQ     10$                     ;BRANCH IF GO IS RESET
017734  005037  001140                  CLR     $GDDAT
017740  104137                          ERROR   137                     ;GO NOT CLEARED BY INIT

017742                          10$:                                    ;END OF TEST

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

```
                                        ;*TEST 27       DIAGNOSTIC MODE TEST
                                        ;;*************************************************************
017742  000004                          TST27:  SCOPE
017744  012737  000027  001226                  MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
017752  000240                                  NOP
017754  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
017762  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
017770  012737  020004  001122                  MOV     #T27,$LPADR     ;LOAD LOOP ON TEST ADDRESS
017776  012737  020004  001124                  MOV     #T27,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
020004                                  T27:
020004  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
020010  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
020014  013701  001456                          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
020020  010037  001136                          MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
020024  062737  000024  001136                  ADD     #RMMR1,$BDADR
020032  005003                                  CLR     R3              ;INITIALIZE ERROR FLAGS
                                        ;INITIALIZE AND VERIFY THAT "DMD" IS RESET
020034  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
020042  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
020046  016037  000024  001142                  MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
020054  042737  177776  001142                  BIC     #↑CDMD,$BDDAT
020062  001403                                  BEQ     10$             ;BRANCH IF "DMD" IS ZERO
020064  005037  001140                          CLR     $GDDAT
020070  104075                                  ERROR   75              ;CANT CLEAR "DMD"
020072                                  10$:
                                        ;SET AND RESET "DMD" USING REGISTER TRANSFER-VERIFY "DMD" NOT S-A-1
020072  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
020100  012760  000000  000024                  MOV     #0,RMMR1(R0)    ;LOAD RMMR1
020106  016037  000024  001142                  MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
020114  042737  177776  001142                  BIC     #↑CDMD,$BDDAT
020122  001405                                  BEQ     20$             ;BRANCH IF DMD NOT S-A-1
020124  005037  001140                          CLR     $GDDAT
020130  104076                                  ERROR   76              ;CANT WRITE DMD=0
020132  052703  000001                          BIS     #BIT0,R3        ;SET ERROR FLAG
020136                                  20$:
                                        ;RESET AND SET "DMD" USING REGISTER TRANSFER-VERIFY "DMD" NOT S-A-0
020136  012760  000000  000024                  MOV     #0,RMMR1(R0)    ;LOAD RMMR1
020144  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
020152  016037  000024  001142                  MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
020160  042737  177776  001142                  BIC     #↑CDMD,$BDDAT
020166  001006                                  BNE     30$             ;BRANCH IF DMD NOT S-A-0
020170  012737  000001  001140                  MOV     #DMD,$GDDAT
020176  104077                                  ERROR   77              ;CAN'T WRITE DMD=1
020200  052703  000002                          BIS     #BIT1,R3        ;SET ERROR FLAG
020204                                  30$:
                                        ;IF NO PREVIOUS ERROR, TEST FOR BIT INTERFERENCE WITH SHIFTING
                                        ;ONE BIT PATTERN
```

```
020204  005703                          TST     R3                      ;ANY ERRORS DETECTED??
020206  001027                          BNE     60$                     ;YES!!
020210  012702  000002                  MOV     #BIT1,R2                ;INITAILIZE DATA PATTERN
020214                          40$:

020214  012760  000000  000024          MOV     #0,RMMR1(R0)            ;LOAD RMMR1

020222  010260  000024                  MOV     R2,RMMR1(R0)            ;LOAD RMMR1

020226  016037  000024  001142          MOV     RMMR1(R0),$BDDAT                ;STORE RMMR1 AT $BDDAT
020234  042737  177776  001142          BIC     #↑CDMD,$BDDAT
020242  001407                          BEQ     50$                     ;BRANCH IF DMD NOT SET
020244  010237  001174                  MOV     R2,$TMPO                ;SAVE DATA
020250  010237  001174                  MOV     R2,$TMPO                ;SAVE DATA
020254  005037  001140                  CLR     $GDDAT                  ;DMD SHOULD BE ZERO
020260  104100                          ERROR   100                     ;DMD SET BY WRONG BIT
020262                          50$:
                                ;       SHIFT TO NEXT DATA BIT AND CONTINUE TEST IF NOT DONE
020262  006302                          ASL     R2
020264  001353                          BNE     40$
020266                          60$:                                    ;END OF TEST
                                ;;*****************************************************************
                                ;*TEST 30        MOL TEST

                                ;;*****************************************************************
020266  000004                  TST30:  SCOPE
020270  012737  000030  001226          MOV     #30,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

020276  000240                          NOP
020300  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
020306  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
020314  012737  020330  001122          MOV     #T30,$LPADR     ;LOAD LOOP ON TEST ADDRESS
020322  012737  020330  001124          MOV     #T30,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
020330                          T30:
020330  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
020334  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
020340  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
020344  012760  000040  000010          MOV     #CLR,RMCS2(R0)          ;CLEAR THE MASSBUS
020352  111160  000010                  MOVB    (R1),RMCS2(R0)          ;SELECT UNIT
020356  010037  001136                  MOV     R0,$BDADR               ;R0=REGISTER ADDRESS
020362  062737  000012  001136          ADD     #RMDS,$BDADR
020370  005003                          CLR     R3                      ;R3=ERROR FLAG
                                ;SET DIAGNOSTIC MODE AND VERIFY THAT "MOL" IS ZERO

020372  012760  000001  000024          MOV     #DMD,RMMR1(R0)          ;LOAD RMMR1

020400  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
020406  042737  167777  001142          BIC     #↑CMOL,$BDDAT
020414  001405                          BEQ     10$
020416  005037  001140                  CLR     $GDDAT
020422  104101                          ERROR   101                     ;"MOL" NOT ZERO
020424  052703  000001                  BIS     #BIT0,R3                ;SET ERROR FLAG
020430                          10$:
                                ;SET MAINTENANCE UNIT READY AND VERIFY THAT "MOL" IS ONE

020430  012760  000001  000024          MOV     #DMD,RMMR1(R0)          ;LOAD RMMR1
```

```
020436  012760  001001  000024            MOV     #DMD!MUR,RMMR1(R0)      ;LOAD RMMR1
020444  016037  000012  001142            MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
020452  042737  167777  001142            BIC     #^CMOL,$BDDAT
020460  001006                            BNE     20$
020462  012737  010000  001140            MOV     #MOL,$GDDAT
020470  104102                            ERROR   102             ;"MOL" NOT ONE
020472  052703  000002                    BIS     #BIT1,R3
020476                            20$:
                                  ;IF NO PREVIOUS ERROR, VERIFY VOLUME VALID IS RESET AND
                                  ;TEST FOR BIT INTERFERENCE
020476  005703                            TST     R3              ;ANY ERROR DETECTED??
020500  001057                            BNE     70$             ;YES!!
020502  016037  000012  001142            MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
020510  042737  177677  001142            BIC     #^CVV,$BDDAT
020516  001403                            BEQ     25$             ;BRANCH IF VV RESET
020520  005037  001140                    CLR     $GDDAT
020524  104135                            ERROR   135             ;VV NOT RESET
020526                            25$:
020526  012702  000001                    MOV     #1,R2           ;INITIALIZE DATA PATTERN
020532                            30$:
020532  012760  000001  000024            MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
020540  010260  000024                    MOV     R2,RMMR1(R0)    ;LOAD RMMR1
020544  016037  000012  001142            MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
020552  042737  167777  001142            BIC     #^CMOL,$BDDAT
020560  005003                            CLR     R3              ;SETUP EXPECTED "MOL"
020562  032702  001000                    BIT     #MUR,R2
020566  001402                            BEQ     40$
020570  052703  010000                    BIS     #MOL,R3         ;"MOL" SHOULD BE ONE
020574  020337  001142            40$:    CMP     R3,$BDDAT       ;IS MOL OK??
020600  001405                            BEQ     50$             ;YES!!
020602  010237  001174                    MOV     R2,$TMP0        ;SAVE TEST PATTERN
020606  010337  001140                    MOV     R3,$GDDAT
020612  104103                            ERROR   103             ;MUR SET BY WRONG BIT
020614                            50$:
                                  ;SHIFT TO NEXT PATTERN
020614  042702  000001                    BIC     #DMD,R2         ;DONT SHIFT DMD
020620  001002                            BNE     60$
020622  012702  000001                    MOV     #DMD,R2         ;DONT TRUNCATE TEST
020626  006302                    60$:    ASL     R2              ;SHIFT TO NEXT BIT
020630  001403                            BEQ     70$             ;BRANCH IF DONE
020632  052702  000001                    BIS     #DMD,R2         ;KEEP DMD ON
020636  000735                            BR      30$             ;CONTINUE
020640                            70$:                            ;END OF TEST

                                  ;;*****************************************************
                                  ;*TEST 31         WRITE LOCK TEST

                                  ;;*****************************************************
020640  000004                    TST31:  SCOPE
```

```
020642  012737  000031  001226          MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
020650  000240                          NOP
020652  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
020660  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
020666  012737  020702  001122          MOV     #T31,$LPADR  ;LOAD LOOP ON TEST ADDRESS
020674  012737  020702  001124          MOV     #T31,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
020702                          T31:
020702  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
020706  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
020712  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
020716  005003                          CLR     R3              ;R3=ERROR FLAG
020720  010037  001136                  MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
020724  062737  000012  001136          ADD     #RMDS,$BDADR
020732  005037  001140                  CLR     $GDDAT          ;WRL SHOULD BE ZERO
020736  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
020744  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                ;SET DIAGNOSTIC MODE AND VERIFY "WRL" IS ZERO
020750  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
020756  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
020764  042737  173777  001142          BIC     #^CWRL,$BDDAT
020772  001403                          BEQ     10$             ;BRANCH IF WRL IS ZERO
020774  104104                          ERROR   104             ;WRL NOT ZERO
020776  052703  000001                  BIS     #BIT0,R3        ;SET ERROR FLAG
021002                          10$:
                                ;SET MAINTENANCE WRITE PROTECT AND VERIFY "WRL" IS ONE
021002  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
021010  012760  000011  000024          MOV     #DMD!MWP,RMMR1(R0)      ;LOAD RMMR1
021016  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
021024  042737  173777  001142          BIC     #^CWRL,$BDDAT
021032  001006                          BNE     20$             ;BRANCH IF WRL IS NOE
021034  012737  004000  001140          MOV     #WRL,$GDDAT     ;WRL SHOULD BE SET
021042  104105                          ERROR   105             ;CANT SET WRL
021044  052703  000002                  BIS     #BIT1,R3        ;SET ERROR FLAG
021050                          20$:
                                ;IF NO PREVIOUS ERROR, TEST FOR BIT INTERFERENCE ON "MWP"
021050  005703                          TST     R3              ;ANY OTHER ERROR??
021052  001045                          BNE     70$             ;YES!!
021054  012702  000001                  MOV     #1,R2           ;INITIALIZE DATA PATTERN
021060                          30$:
                                ;       TRANSFER DATA TO RMMR1, READ RMDS AND VERIFY WRL
021060  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
021066  010260  000024                  MOV     R2,RMMR1(R0)    ;LOAD RMMR1
021072  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
021100  042737  173777  001142          BIC     #^CWRL,$BDDAT   ;CLEARUP RECEIVED "WRL"
021106  005003                          CLR     R3              ;GENERATE EXPECTED "WRL"
021110  032702  000010                  BIT     #MWP,R2
```

```
        021114  001402                          BEQ     40$
        021116  052703  004000                  BIS     #WRL,R3         ;WRL SHOULD BE SET
        021122  020337  001142          40$:    CMP     R3,$BDDAT       ;IS WRL OK??
        021126  001405                          BEQ     50$             ;YES!!
        021130  010337  001140                  MOV     R3,$GDDAT       ;SAVE EXPECTED
        021134  010237  001174                  MOV     R2,$TMP0        ;SAVE DATA PATTERN
        021140  104106                          ERROR   106             ;BIT INTERFERENCE WITH MWP
        021142                          50$:

                                        ;       ADVANCE TO NEXT DATA BIT
        021142  042702  000001                  BIC     #DMD,R2         ;DONT SHIFT DMD
        021146  001002                          BNE     60$
        021150  012702  000001                  MOV     #DMD,R2         ;DONT TRUNCATE TEST
        021154  006302                  60$:    ASL     R2              ;SHIFT DATA BIT
        021156  001403                          BEQ     70$             ;EXIT IF DONE
        021160  052702  000001                  BIS     #DMD,R2         ;KEEP DIAGNOSTIC MODE ON
        021164  000735                          BR      30$             ;CONTINUE TEST

        021166                          70$:                            ;END OF TEST

                                        ;;*****************************************************************
                                        ;*TEST 32       DRIVE FAULT TEST

                                        ;;*****************************************************************
        021166  000004                  TST32:  SCOPE
        021170  012737  000032  001226          MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

        021176  000240                          NOP
        021200  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
        021206  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
        021214  012737  021230  001122          MOV     #T32,$LPADR ;LOAD LOOP ON TEST ADDRESS
        021222  012737  021230  001124          MOV     #T32,$LPERR ;LOAD LOOP ON ERROR ADDRESS
        021230                          T32:
        021230  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
        021234  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
        021240  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
        021244  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
        021252  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
        021256  005003                          CLR     R3              ;INITIALIZE ERROR FLAGS
        021260  010037  001136                  MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
        021264  062737  000042  001136          ADD     #RMER2,$BDADR
        021272  005037  001140                  CLR     $GDDAT          ;"DVC" AND "UNS" SHOULD BE ZERO
                                        ;SET AND RESET MAINTENANCE DRIVE FAULT, VERIFY THAT "DVC" IS NOT
                                        ;STUCK-AT-ONE.

        021276  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

        021304  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

        021312  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

        021320  016037  000042  001142          MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
        021326  042737  177577  001142          BIC     #^CDVC,$BDDAT   ;IS "DVC" RESET??
        021334  001406                          BEQ     10$             ;YES!!
        021336  104107                          ERROR   107             ;CANT RESET DVC
        021340  052703  000001                  BIS     #BIT0,R3        ;SET ERROR FLAG
        021344  012737  040000  001140          MOV     #UNS,$GDDAT
```

```
021352                              10$:

                                    ;VERIFY THAT "UNS" IS SAME AS "DVC"

021352  016037  000014  001142              MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
021360  042737  137777  001142              BIC     #↑CUNS,$BDDAT
021366  023737  001140  001142              CMP     $GDDAT,$BDDAT   ;IS "UNS" OK??
021374  001414                              BEQ     30$             ;YES!!
021376  010037  001136                      MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
021402  062737  000014  001136              ADD     #RMER1,$BDADR
021410  032737  040000  001140              BIT     #UNS,$GDDAT     ;SHOULD "UNS" BE ON??
021416  001002                              BNE     20$             ;YES!!
021420  104110                              ERROR   110             ;UNS IS SET, DVS IS RESET
021422  000401                              BR      30$
021424                              20$:
021424  104111                              ERROR   111             ;UNS IS RESET, DVC IS SET
021426                              30$:

                                    ;RESET AND SET "MDF", VERIFY THAT "DVC" IS NOT S-A-0.
021426  012737  000200  001140              MOV     #DVC,$GDDAT     ;DVC SHOULD BE ON

021434  012760  000001  000024              MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

021442  012760  000101  000024              MOV     #DMD!MDF,RMMR1(R0)      ;LOAD RMMR1

021450  016037  000042  001142              MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
021456  042737  177577  001142              BIC     #↑CDVC,$BDDAT
021464  001012                              BNE     40$             ;BRANCH IF DVC IS SET
021466  010037  001136                      MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
021472  062737  000042  001136              ADD     #RMER2,$BDADR
021500  104112                              ERROR   112             ;CANT SET DVC
021502  052703  000002                      BIS     #BIT1,R3        ;SET ERROR FLAG
021506  005037  001140                      CLR     $GDDAT          ;UNS SHOULD BE OFF
021512                              40$:

                                    ;VERIFY THAT "UNS" IS SAME AS "DVC"
021512  005737  001140                      TST     $GDDAT          ;CHANGE DVC TO UNS
021516  001403                              BEQ     50$
021520  012737  040000  001140              MOV     #UNS,$GDDAT
021526                              50$:

021526  016037  000014  001142              MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
021534  042737  137777  001142              BIC     #↑CUNS,$BDDAT
021542  023737  001140  001142              CMP     $GDDAT,$BDDAT
021550  001414                              BEQ     70$             ;BRANCH IF UNS IS OK
021552  010037  001136                      MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
021556  062737  000014  001136              ADD     #RMER1,$BDADR
021564  032737  040000  001140              BIT     #UNS,$GDDAT     ;SHOULD UNS BE ON??
021572  001002                              BNE     60$             ;YES!!
021574  104113                              ERROR   113             ;UNS IS SET, DVC IS RESET
021576  000401                              BR      70$
021600                              60$:
021600  104114                              ERROR   114             ;UNS IS RESET, DVC IS SET
021602                              70$:

                                    ;IF THERE WERE NO PREVIOUS ERRORS, TEST FOR BIT INTERFERENCE ON
```

```
                                        ;"MDF"
021602  005703                                  TST     R3
021604  001056                                  BNE     120$                    ;BRANCH IF ANY OTHER ERRORS
021606  012702  000001                          MOV     #1,R2                   ;INITIALIZE TEST PATTERN
021612                                  80$:

021612  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

021620  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

021626  010260  000024                          MOV     R2,RMMR1(R0)    ;LOAD RMMR1

021632  016037  000042  001142                  MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
021640  042737  177577  001142                  BIC     #↑CDVC,$BDDAT   ;GET RESULTS
021646  005037  001140                          CLR     $GDDAT          ;SETUP EXPECTED
021652  032702  000100                          BIT     #MDF,R2         ;WAS MDF SET??
021656  001403                                  BEQ     90$             ;NO!!
021660  052737  000200  001140                  BIS     #DVC,$GDDAT     ;YES-DVC SHOULD BE ON
021666  023737  001140  001142          90$:    CMP     $GDDAT,$BDDAT
021674  001410                                  BEQ     100$            ;BRANCH IF DVC IS OK
021676  010037  001136                          MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
021702  062737  000042  001136                  ADD     #RMER2,$BDADR
021710  010237  001174                          MOV     R2,$TMP0        ;SAVE TEST PATTERN
021714  104115                                  ERROR   115             ;MDF SET BY WRONG BIT
021716                                  100$:
                                        ;SHIFT TO NEXT BIT POSITION
021716  042702  000001                          BIC     #DMD,R2         ;DONT SHIFT DMD
021722  001002                                  BNE     110$
021724  012702  000001                          MOV     #DMD,R2         ;DONT TRUNCATE TEST
021730  006302                          110$:   ASL     R2              ;SHIFT
021732  001403                                  BEQ     120$            ;EXIT IF DONE
021734  052702  000001                          BIS     #DMD,R2         ;KEEP DMD ON
021740  000724                                  BR      80$             ;CONTINUE

021742                                  120$:                           ;END OF TEST


                                        ;;*****************************************************
                                        ;*TEST 33        SEEK ERROR TEST

                                        ;;*****************************************************
021742  000004                          TST33:  SCOPE
021744  012737  000033  001226                  MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

021752  000240                                  NOP
021754  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
021762  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
021770  012737  022004  001122                  MOV     #T33,$LPADR     ;LOAD LOOP ON TEST ADDRESS
021776  012737  022004  001124                  MOV     #T33,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
022004                                  T33:
022004  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
022010  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
022014  013701  001456                          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
022020  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
022026  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
022032  005003                                  CLR     R3              ;CLEAR ERROR FLAGS
022034  010037  001136                          MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
```

```
022040 062737 000042 001136          ADD     #RMER2,$BDADR
                                     ;SET DIAGNOSTIC MODE AND VERIFY THAT "SKI" CAN BE RESET
022046 012760 000001 000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
022054 012760 000000 000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
022062 016037 000042 001142          MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
022070 042737 137777 001142          BIC     #↑CSKI,$BDDAT
022076 001405                        BEQ     10$             ;BRANCH IF SKI IS RESET
022100 005037 001140                 CLR     $GDDAT          ;SKI SHOULD BE ZERO
022104 104116                        ERROR   116             ;CANT RESET "SKI"
022106 052703 000001                 BIS     #BIT0,R3        ;SET ERROR FLAG
022112                        10$:
                                     ;SET MAINTENANCE SEEK ERROR AND VERIFY THAT "SKI" CAN BE SET
022112 012760 000001 000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
022120 012760 000000 000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
022126 012760 000201 000024          MOV     #DMD!MSER,RMMR1(R0)     ;LOAD RMMR1
022134 016037 000042 001142          MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
022142 042737 137777 001142          BIC     #↑CSKI,$BDDAT
022150 001005                        BNE     20$             ;BRANCH IF SKI IS SET
022152 012737 040000 001140          MOV     #SKI,$GDDAT     ;CANT SET SKI
022160 052703 000002                 BIS     #BIT1,R3        ;SET ERROR FLAG
022164                        20$:
                                     ;IF NO PREVIOUS ERROR, CHECK FOR BIT INTERFERENCE SETTING MAINTENANCE
                                     ;SEEK ERROR.
022164 005703                        TST     R3
022166 001051                        BNE     70$             ;BRANCH IF ANY OTHER ERRORS
022170 012702 000001                 MOV     #1,R2           ;INITIALIZE TEST PATTERN
022174                        30$:
022174 012760 000001 000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
022202 012760 000000 000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
022210 010260 000024                 MOV     R2,RMMR1(R0)    ;LOAD RMMR1
022214 016037 000042 001142          MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
022222 042737 137777 001142          BIC     #↑CSKI,$BDDAT   ;GET SKI STATUS
022230 005037 001140                 CLR     $GDDAT          ;SETUP EXPECTED RESULT
022234 032702 000200                 BIT     #MSER,R2
022240 001403                        BEQ     40$
022242 052737 040000 001140          BIS     #SKI,$GDDAT     ;SKI SHOULD BE ON
022250 023737 001140 001142   40$:   CMP     $GDDAT,$BDDAT
022256 001403                        BEQ     50$             ;BRANCH IF SKI IS OK
022260 010237 001174                 MOV     R2,$TMP0        ;SAVE TEST PATTERN
022264 104120                        ERROR   120             ;SKI SET BY WRONG BIT
022266                        50$:
                              ;      ADVANCE TEST PATTERN IN R2
```

```
022266  042702  000001                  BIC     #DMD,R2            ;DONT SHIFT DMD BIT
022272  001002                          BNE     60$
022274  012702  000001                  MOV     #DMD,R2            ;DONT TRUNCATE TEST
022300  006302                  60$:    ASL     R2                 ;SHIFT TO NEXT BIT
022302  001403                          BEQ     70$                ;EXIT IF DONE
022304  052702  000001                  BIS     #DMD,R2            ;KEEP DMD ON
022310  000731                          BR      30$                ;CONTINUE TEST

022312                          70$:                               ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 34        PIP TEST

                                ;;*****************************************************************
022312  000004                  TST34:  SCOPE
022314  012737  000034  001226          MOV     #34,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX

022322  000240                          NOP
022324  012737  000024  001120          MOV     #20.,$ICNT         ;20 ITERATIONS
022332  112737  000001  001131          MOVB    #1,$ERMAX          ;ONE ERROR ALLOWED
022340  012737  022354  001122          MOV     #T34,$LPADR  ;LOAD LOOP ON TEST ADDRESS
022346  012737  022354  001124          MOV     #T34,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
022354                          T34:
022354  012706  001100                  MOV     #STACK,SP          ;LOAD THE STACK POINTER
022360  013700  001276                  MOV     $BASE,R0           ;R0 = UNIBUS ADDRESS OF UUT
022364  013701  001456                  MOV     TSTQUE,R1          ;R1 = POINTER TO DEVICE
022370  012760  000040  000010          MOV     #CLR,RMCS2(R0)     ;CLEAR THE MASSBUS
022376  111160  000010                  MOVB    (R1),RMCS2(R0)     ;SELECT UNIT
022402  005003                          CLR     R3                 ;RESET ERROR FLAGS
022404  010037  001136                  MOV     R0,$BDADR          ;SETUP REGISTER ADDRESS
022410  062737  000012  001136          ADD     #RMDS,$BDADR

                                ;SET MAINTENANCE ON CYLINDER "MOC" AND VERIFY THAT "PIP" CAN BE RESET.

022416  012760  000001  000024          MOV     #DMD,RMMR1(R0)     ;LOAD RMMR1

022424  012760  000401  000024          MOV     #DMD!MOC,RMMR1(R0)         ;LOAD RMMR1

022432  016037  000012  001142          MOV     RMDS(R0),$BDDAT    ;STORE RMDS AT $BDDAT
022440  042737  157777  001142          BIC     #^CPIP,$BDDAT
022446  001405                          BEQ     10$                ;BRANCH IF PIP IS RESET
022450  005037  001140                  CLR     $GDDAT
022454  104121                          ERROR   121                ;CANT RESET PIP
022456  052703  000001                  BIS     #BIT0,R3           ;SET ERROR FLAG
022462                          10$:
                                ;RESET MAINTENANCE ON CYLINDER AND VERIFY THAT "PIP" CAN BE SET

022462  012760  000001  000024          MOV     #DMD,RMMR1(R0)     ;LOAD RMMR1

022470  016037  000012  001142          MOV     RMDS(R0),$BDDAT    ;STORE RMDS AT $BDDAT
022476  042737  157777  001142          BIC     #^CPIP,$BDDAT
022504  001006                          BNE     20$                ;BRANCH IF PIP IS SET
022506  012737  020000  001140          MOV     #PIP,$GDDAT
022514  104122                          ERROR   122                ;CANT SET PIP
022516  052703  000002                  BIS     #BIT1,R3           ;SET ERROR FLAG
022522                          20$:
```

```
                                        ;IF NO PREVIOUS ERROR, TEST FOR ADJACENT BIT SETTING "MOC"
022522  005703                                  TST     R3
022524  001046                                  BNE     70$                     ;BRANCH IF ANY PREVIOUS ERROR
022526  012702  000001                          MOV     #1,R2                   ;INITIALIZE TEST PATTERN
022532                                  30$:
                                        ;       WRITE THE TEST PATTERN, CHECK MOC USING PIP

022532  012760  000001  000024                  MOV     #DMD,RMMR1(R0)          ;LOAD RMMR1

022540  010260  000024                          MOV     R2,RMMR1(R0)            ;LOAD RMMR1

022544  016037  000012  001142                  MOV     RMDS(R0),$BDDAT         ;STORE RMDS AT $BDDAT
022552  042737  157777  001142                  BIC     #^CPIP,$BDDAT           ;GET PIP STATUS
022560  005037  001140                          CLR     $GDDAT                  ;SETUP EXPECTED RESULT
022564  032702  000400                          BIT     #MOC,R2
022570  001003                                  BNE     40$
022572  052737  020000  001140                  BIS     #PIP,$GDDAT             ;PIP SHOULD BE SET
022600  023737  001140  001142          40$:    CMP     $GDDAT,$BDDAT
022606  001403                                  BEQ     50$                     ;BRANCH IF PIP OK
022610  010237  001174                          MOV     R2,$TMP0                ;SAVE TEST PATTERN
022614  104123                                  ERROR   123                     ;MOC SET BY WRONG BIT
022616                                  50$:
                                        ;       ADVANCE THE TEST PATTERN
022616  042702  000001                          BIC     #DMD,R2                 ;DONT SHIFT DMD
022622  001002                                  BNE     60$
022624  012702  000001                          MOV     #DMD,R2                 ;DONT TRUNCATE TEST
022630  006302                          60$:    ASL     R2                      ;SHIFT BIT
022632  001403                                  BEQ     70$                     ;EXIT IF DONE
022634  052702  000001                          BIS     #DMD,R2                 ;KEEP DMD ON
022640  000734                                  BR      30$                     ;CONTINUE TEST

022642                                  70$:                                    ;END OF TEST


                                        ;;*****************************************************************
                                        ;*TEST 35               EBL TEST

                                        ;;*****************************************************************
022642  000004                          TST35:  SCOPE
022644  012737  000035  001226                  MOV     #35,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

022652  000240                                  NOP
022654  012737  000024  001120                  MOV     #20.,$ICNT              ;20 ITERATIONS
022662  112737  000001  001131                  MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
022670  012737  022704  001122                  MOV     #T35,$LPADR     ;LOAD LOOP ON TEST ADDRESS
022676  012737  022704  001124                  MOV     #T35,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
022704                                  T35:
022704  012706  001100                          MOV     #STACK,SP               ;LOAD THE STACK POINTER
022710  013700  001276                          MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
022714  013701  001456                          MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
022720  005003                                  CLR     R3                      ;RESET ERROR FLAGS
022722  010037  001136                          MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS
022726  062737  000024  001136                  ADD     #RMMR1,$BDADR
022734  005037  001140                          CLR     $GDDAT                  ;SETUP EXPECTED RESULT
                                        ;CLEAR AND VERIFY THAT END OF BLOCK IS RESET
022740  012760  000040  000010                  MOV     #CLR,RMCS2(R0)          ;CLEAR THE MASSBUS
022746  111160  000010                          MOVB    (R1),RMCS2(R0)          ;SELECT UNIT
```

```
022752  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
022760  042737  157777  001142          BIC     #^CEBL,$BDDAT
022766  001403                          BEQ     10$                     ;BRANCH IF EBL IS RESET
022770  104124                          ERROR   124                     ;CANT CLEAR EBL
022772  052703  000001                  BIS     #BIT0,R3                ;SET ERROR FLAG
022776                          10$:
                                ;SET AND RESET DIAGNOSTIC END OF BLOCK, CHECK FOR EBL S-A-1.

022776  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

023004  012760  020001  000024          MOV     #DMD!DEBL,RMMR1(R0)     ;LOAD RMMR1

023012  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

023020  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
023026  042737  157777  001142          BIC     #^CEBL,$BDDAT
023034  001403                          BEQ     20$                     ;BRANCH IF EBL IS RESET
023036  104125                          ERROR   125                     ;DEBL APPEARS S-A-1
023040  052703  000001                  BIS     #BIT0,R3                ;SET ERROR FLAG
023044                          20$:
                                ;RESET AND SET DIAGNOSTIC END OF BLOCK, CHECK FOR EBL S-A-0

023044  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

023052  012760  020001  000024          MOV     #DMD!DEBL,RMMR1(R0)     ;LOAD RMMR1

023060  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
023066  042737  157777  001142          BIC     #^CEBL,$BDDAT
023074  001006                          BNE     30$                     ;BRANCH IF EBL IS SET
023076  012737  020000  001140          MOV     #EBL,$GDDAT
023104  104126                          ERROR   126                     ;CANT SET EBL
023106  052703  000002                  BIS     #BIT1,R3                ;SET ERROR FLAG
023112                          30$:
                                ;IF NO PREVIOUS ERRORS, TEST FOR ADJACENT BIT INTERFERENCE ON "DEBL".
023112  005703                          TST     R3
023114  001042                          BNE     70$                     ;BRANCH IF ANY ERROR
023116  012702  000001                  MOV     #1,R2                   ;INITIALIZE TEST PATTERN
023122                          40$:
                                ;       WRITE, READ AND VERIFY THE TEST PATTERN IN R2

023122  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

023130  010260  000024                  MOV     R2,RMMR1(R0)    ;LOAD RMMR1

023134  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
023142  042737  157777  001142          BIC     #^CEBL,$BDDAT
023150  010203                          MOV     R2,R3                   ;GENERATE EXPECTED RESULT
023152  042703  157777                  BIC     #^CEBL,R3
023156  020337  001142                  CMP     R3,$BDDAT
023162  001405                          BEQ     50$                     ;BRANCH IF EBL IS OK
023164  010337  001140                  MOV     R3,$GDDAT               ;SAVE EXPECTED RESULT
023170  010237  001174                  MOV     R2,$TMP0                ;SAVE TEST PATTERN
023174  104127                          ERROR   127                     ;DEBL SET(RESET) BY WRONG BIT
023176                          50$:
                                ;       SHIFT TO NEXT BIT POSITION
```

```
023176  042702  000001                  BIC     #DMD,R2         ;DONT SHIFT DMD
023202  001002                          BNE     60$
023204  012702  000001                  MOV     #DMD,R2         ;DONT TRUNCATE DMD
023210  006302                  60$:    ASL     R2              ;SHIFT TO NEXT BIT
023212  001403                          BEQ     70$             ;EXIT IF DONE
023214  052702  000001                  BIS     #DMD,R2         ;KEEP DMD ON
023220  000740                          BR      40$             ;CONTINUE TEST

023222                          70$:                            ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 36        LAST SECTOR, LAST TRACK TEST

                                ;;*****************************************************************
023222  000004                  TST36:  SCOPE
023224  012737  000036  001226          MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
023232  000240                          NOP
023234  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
023242  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
023250  012737  023264  001122          MOV     #T36,$LPADR     ;LOAD LOOP ON TEST ADDRESS
023256  012737  023264  001124          MOV     #T36,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
023264                          T36:
023264  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
023270  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
023274  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
023300  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
023306  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
023312  005002                          CLR     R2              ;INITIALIZE TEST PATTERN
023314  010037  001136                  MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
023320  062737  000024  001136          ADD     #RMMR1,$BDADR
023326  005037  001434                  CLR     RMOFO           ;START IN 18 BIT MODE
023332                          10$:
                                ;TRANSFER TEST PATTERN TO RMDA THEN VERIFY LAST SECTOR "LS" AND LAST
                                ;SECTOR/TRACK "LST" FOR EACH TRANSFER. THE TABLE BELOW LISTS THE VALUE
                                ;OF RMDA FOR WHICH LS AND LST ARE SET.
                                ;
                                ;       18 BIT MODE     16 BIT MODE
                                ;LS=    XXX035          XXX037
                                ;
                                ;LST=   002035          002037


023332  013760  001434  000032          MOV     RMOFO,RMOF(R0)  ;LOAD RMOF

023340  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA

023344  016037  000024  001352          MOV     RMMR1(R0),RMMR1I        ;STORE RMMR1 IN INPUT BUFFER
023352  013737  001352  001142          MOV     RMMR1I,$BDDAT   ;VERIFY "LS"
023360  042737  177773  001142          BIC     #^CLS,$BDDAT
023366  005037  001140                  CLR     $GDDAT          ;GENERATE EXPECTED "LS"
023372  032737  010000  001434          BIT     #FMT16,RMOFO    ;16 BIT MODE??
023400  001007                          BNE     20$             ;YES!!
023402  122702  000035                  CMPB    #035,R2
023406  001012                          BNE     30$
023410  052737  000004  001140          BIS     #LS,$GDDAT      ;LS SHOULD BE ON-18 BIT MODE
023416  000406                          BR      30$
```

```
023420  122702  000037          20$:    CMPB    #037,R2
023424  001003                          BNE     30$
023426  052737  000004  001140          BIS     #LS,$GDDAT      ;LS SHOULD BE ON-16 BIT MODE
023434  023737  001140  001142  30$:    CMP     $GDDAT,$BDDAT
023442  001407                          BEQ     40$             ;BRANCH IF LS IS CORRECT
023444  010237  001174                  MOV     R2,$TMP0        ;SAVE TEST PATTERN
023450  013737  001434  001176          MOV     RMOFO,$TMP1     ;SAVE OFFSET REGISTER FOR ERROR
023456  104130                          ERROR   130             ;LS NOT CORRECT FOR RMDA
023460  000453                          BR      80$             ;SKIP TO NEXT
023462  013737  001352  001142  40$:    MOV     RMMR1I,$BDDAT   ;VERIFY "LST"
023470  042737  177775  001142          BIC     #^CLST,$BDDAT
023476  005037  001140                  CLR     $GDDAT          ;GENERATE EXPECTED "LST"
023502  032737  010000  001434          BIT     #FMT16,RMOFO    ;16 BIT MODE??
023510  001007                          BNE     50$             ;YES!!
023512  022702  002035                  CMP     #002035,R2
023516  001012                          BNE     60$
023520  052737  000002  001140          BIS     #LST,$GDDAT     ;LST SHOULD BE ON-18 BIT MODE
023526  000406                          BR      60$
023530  022702  002037          50$:    CMP     #002037,R2
023534  001003                          BNE     60$
023536  052737  000002  001140          BIS     #LST,$GDDAT     ;LST SHOULD BE ON-16 BIT MODE
023544  023737  001140  001142  60$:    CMP     $GDDAT,$BDDAT
023552  001404                          BEQ     70$
023554  010237  001174                  MOV     R2,$TMP0        ;SAVE TEST PATTERN
023560  104131                          ERROR   131             ;LST NOT CORRECT FOR RMDA
023562  000412                          BR      80$             ;SKIP TO NEXT

023564                          70$:
                                ;       ADVANCE TO NEXT TEST PATTERN, CHANGE TO 16 BIT MODE IF ALL
                                ;18 BIT TESTS DONE.
023564  005202                          INC     R2              ;INCREMENT PATTERN
023566  001261                          BNE     10$             ;CONTINUE IF NOT DONE
023570  032737  010000  001434          BIT     #FMT16,RMOFO    ;DONE 16 BIT TOO??
023576  001004                          BNE     80$             ;YES!!
023600  012737  010000  001434          MOV     #FMT16,RMOFO    ;DO 16 BIT FORMAT TEST
023606  000651                          BR      10$

023610                          80$:                            ;END OF TEST

                                ;;*************************************************************
                                ;*TEST 37          RMDA COUNT TEST

                                ;;*************************************************************
023610  000004                  TST37:  SCOPE
023612  012737  000037  001226          MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

023620  000240                          NOP
023622  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
023630  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
023636  012737  023652  001122          MOV     #T37,$LPADR     ;LOAD LOOP ON TEST ADDRESS
023644  012737  023652  001124          MOV     #T37,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
023652                          T37:
023652  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
023656  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
023662  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
023666  010037  001136                  MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
```

```
 7766  023672  062737  000006  001136          ADD     #RMDA,$BDADR
 7767  023700  005037  001434                  CLR     RMOFO              ;START WITH 18 BIT FORMAT
 7768  023704  012737  000035  024326          MOV     #035,110$          ;SETUP LAST SECTOR FOR 18 BIT
 7769  023712  012737  000001  001140          MOV     #1,$GDDAT          ;SETUP FIRST COUNT
 7770                                  ;;*****************************************************************
```

```
 7771                                  ;INCREMENT SECTOR COUNT USING DIAGNOSTIC END OF BLOCK STARTING AT
 7772                                  ;SECTOR 0 AND CONTINUING UNTIL TRACK ADDRESS INCREMENTS
```

```
023720                                  10$:
                                        ;       .CLEAR THE MASSBUS
                                        ;       .SET FORMAT
                                        ;       .LOAD SECTOR AND TRACK ADDRESS
                                        ;       .ENABLE DEBUG CLOCK
                                        ;       .SET GO BIT
023720  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
023726  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

023732  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

023740  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1

023746  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

023754  012760  000000  000006                  MOV     #0,RMDA(R0)     ;LOAD RMDA

023762  013760  001434  000032                  MOV     RMOF0,RMOF(R0)  ;LOAD RMOF
023770  013737  001434  001174                  MOV     RMOF0,$TMP0     ;SAVE OFFSET FOR ERROR TYPE

023776  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1

024004  012760  000001  000000                  MOV     #GO,RMCS1(R0)   ;LOAD RMCS1
```

```
024012                          25$:

                                ;       SET AND RESET EBL TO INCREMENT RMDA THEN VERIFY RMDA.

024012  012760  060001  000024          MOV     #DMD!DBEN!DEBL,RMMR1(R0)        ;LOAD RMMR1

024020  012760  040001  000024          MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1

024026  016037  000006  001142          MOV     RMDA(R0),$BDDAT ;STORE RMDA AT $BDDAT
024034  023737  001142  001140          CMP     $BDDAT,$GDDAT
024042  001402                          BEQ     30$             ;BRANCH IF RMDA OK
024044  104132                          ERROR   132             ;SECTOR COUNT NOT INCREMENTING
024046  000416                          BR      50$             ;OUT OF SYNC-SKIP TO NEXT
024050                          30$:
                                ;       ADVANCE EXPECTED SECTOR COUNT AND CONTINUE IF ONE CYCLE NOT
                                ;COMPLETE
024050  005237  001140                  INC     $GDDAT          ;INCREMENT EXPECTED SECTOR
024054  123737  001142  024326          CMPB    $BDDAT,110$     ;WAS THE LAST SECTOR JUST COUNTED??
024062  001004                          BNE     40$             ;NO!!
024064  105037  001140                  CLRB    $GDDAT          ;YES-NEXT SECTOR SHOULD BE ZERO
024070  105237  001141                  INCB    $GDDAT+1        ;INCREMENT TRACK ADDRESS
024074  105737  001142          40$:    TSTB    $BDDAT          ;HAS A FULL CYCLE BEEN COUNTED??
024100  001401                          BEQ     50$             ;YES-DO NEXT
024102  000743                          BR      25$             ;CONTINUE SECTOR TEST

024104                          50$:
                                ;;*************************************************************
                                ;INCREMENT TRACK COUNT USING DIAGNOSTIC END OF BLOCK.  START AT TRACK 0,
                                ;LAST SECTOR AND COUNT ONE COMPLETE TRACK CYCLE.
024104  013737  024326  001410          MOV     110$,RMDA0      ;STARTING TRACK ADDRESS
024112  012737  000400  001140          MOV     #000400,$GDDAT  ;FIRST VALUE AFTER INCREMENT

024120                          60$:

                                ;       .CLEAR THE MASSBUS
                                ;       .SET FORMAT
                                ;       .LOAD LAST SECTOR ADDRESS AND TEST TRACK ADDRESS
                                ;       .ENABLE DEBUG CLOCK
                                ;       .SET GO BIT

024120  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
024126  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

024132  013760  001434  000032          MOV     RMOF0,RMOF(R0)  ;LOAD RMOF

024140  013760  001410  000006          MOV     RMDA0,RMDA(R0)  ;LOAD RMDA

024146  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

024154  012760  040001  000024          MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1

024162  012760  000001  000000          MOV     #GO,RMCS1(R0)   ;LOAD RMCS1
                                ;CLOCK RMDA USING DIAGNOSTIC END OF BLOCK

024170  012760  060001  000024          MOV     #DMD!DBEN!DEBL,RMMR1(R0)        ;LOAD RMMR1

```

```
024176  012760  040001  000024           MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1
                                 ;       VERIFY RMDA ACCORDING TO $GDDAT

024204  016037  000006  001142           MOV     RMDA(R0),$BDDAT ;STORE RMDA AT $BDDAT
024212  023737  001140  001142           CMP     $GDDAT,$BDDAT
024220  001402                           BEQ     70$
024222  104133                           ERROR   133             ;TRACK COUNT NOT INCREMENTING
024224  000420                           BR      90$             ;OUT OF SYNC-SKIP TO NEXT
024226                           70$:
                                 ;       SETUP FOR NEXT INCREMENT OF RMDA TRACK ADDRESS
024226  105237  001141                   INCB    $GDDAT+1        ;ADVANCE EXPECTED TRACK
024232  123727  001143  000004           CMPB    $BDDAT+1,#4     ;WAS THE LAST TRACK JUST COUNTED??
024240  001002                           BNE     80$             ;NO!!
024242  005037  001140                   CLR     $GDDAT          ;YES-NEXT TRACK, SECTOR SHOULD BE ZERO
024246  013737  001142  001410   80$:    MOV     $BDDAT,RMDAO    ;HAS A FULL CYCLE BEEN COUNTED??
024254  001404                           BEQ     90$             ;YES!!
024256  113737  024326  001410           MOVB    110$,RMDAO      ;INCREMENT FROM LAST SECTOR
024264  000715                           BR      60$
024266                           90$:
                                 ;;*******************************************************
                                 ;REPEAT THE TEST IN 16 BIT FORAT
024266  032737  010000  001434           BIT     #FMT16,RMOFO    ;DONE BOTH FORMATS??
024274  001013                           BNE     100$            ;YES!!
024276  012737  010000  001434           MOV     #FMT16,RMOFO    ;SET FORMAT BIT FOR 16
024304  012737  000037  024326           MOV     #037,110$       ;SET LAST SECTOR FOR 16
024312  012737  000001  001140           MOV     #1,$GDDAT       ;SET FIRST COUNT VALUE
024320  000137  023720                   JMP     10$             ;REPEAT TEST

024324  000401                   100$:   BR      120$

024326  000000                   110$:   .WORD                   ;STORAGE FOR LAST SECTOR VALUE

024330                           120$:                           ;END OF TEST
```

```
                                        ;;*******************************************************
                                        ;*TEST 40        RMDC COUNT TEST
                                        
                                        ;;*******************************************************
024330  000004                          TST40:  SCOPE
024332  012737  000040  001226                  MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

024340  000240                                  NOP
024342  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
024350  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
024356  012737  024372  001122                  MOV     #T40,$LPADR  ;LOAD LOOP ON TEST ADDRESS
024364  012737  024372  001124                  MOV     #T40,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
024372                                  T40:
024372  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
024376  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
024402  013701  001456                          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
024406  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
024414  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                        ;CLEAR RMDC, SET FORMAT AND SETUP PROGRAM PARAMETERS
024420  010037  001136                          MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
024424  062737  000034  001136                  ADD     #RMDC,$BDADR
024432  005037  001434                          CLR     RMOFO           ;START WITH 18 BIT FORMAT
024436  012737  002035  001410                  MOV     #002035,RMDAO   ;LOAD LAST SECTOR/TRACK VALUE
024444  012737  000001  001140          10$:    MOV     #1,$GDDAT       ;LOAD FIRST INCREMENTAL VALUE

024452  012760  000000  000034                  MOV     #0,RMDC(R0)     ;LOAD RMDC

024460  013760  001434  000032                  MOV     RMOFO,RMOF(R0)  ;LOAD RMOF
024466  013737  001434  001174                  MOV     RMOFO,$TMPO     ;SAVE FOR ERROR MSG
024474                                  20$:

                                        ;       .CLEAR THE MASSBUS
                                        ;       .SET OFFSET
                                        ;       .LOAD LAST SECTOR AND TRACK ADDRESS
                                        ;       .ENABLE DEBUG CLOCK
                                        ;       .SET GO BIT

024474  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
024502  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

024506  013760  001434  000032                  MOV     RMOFO,RMOF(R0)  ;LOAD RMOF

024514  013760  001410  000006                  MOV     RMDAO,RMDA(R0)  ;LOAD RMDA

024522  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

024530  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1

024536  012760  000001  000000                  MOV     #GO,RMCS1(R0)   ;LOAD RMCS1
                                        ;CLOCK THE CYLINDER ADDRESS USING DEBL

024544  012760  060001  000024                  MOV     #DMD!DBEN!DEBL,RMMR1(R0)        ;LOAD RMMR1

024552  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1
```

```

024560	016037	000034	001142			MOV	RMDC(R0),$BDDAT	;STORE RMDC AT $BDDAT
024566	023737	001140	001142			CMP	$GDDAT,$BDDAT
024574	001402					BEQ	30$		;BRANCH IF RMDC=RMDC+1
024576	104140					ERROR	140		;CANT INCREMENT RMDC
024600	000427					BR	60$		;OUT OF SYNC-SKIP TO END
024602					30$:
					;ADVANCE EXPECTED RESULT FOR NEXT INCREMENT
024602	005237	001140				INC	$GDDAT		;ADVANCE NEXT RESULT
024606	022737	002000	001140			CMP	#1777+1,$GDDAT	;SHOULD NEXT VALUE BE ZERO??
024614	001002					BNE	40$		;NO!!
024616	005037	001140				CLR	$GDDAT		;YES-RMDC SHOULD OVERFLOW
024622	005737	001142			40$:	TST	$BDDAT		;IS ONE CYCLE COMPLETE??
024626	001401					BEQ	50$		;YES!!
024630	000721					BR	20$		;CONTINUE
024632					50$:
					;REPEAT TEST IN 16 BIT MODE-ELSE DONE
024632	032737	010000	001434			BIT	#FMT16,RMOFO	;DONE BOTH MODES??
024640	001007					BNE	60$		;YES!!
024642	012737	010000	001434			MOV	#FMT16,RMOFO	;SET 16 BIT FORMAT
024650	012737	002037	001410			MOV	#002037,RMDAO	;LOAD LAST SECTOR/TRACK VALUE
024656	000672					BR	10$		;REPEAT TEST

024660					60$:				;END OF TEST

					;;*****************************************************************
					;*TEST 41		LBT TEST

					;;*****************************************************************
024660	000004				TST41:	SCOPE
024662	012737	000041	001226			MOV	#41,$TESTN	;;SET TEST NUMBER IN APT MAIL BOX

024670	000240					NOP
024672	012737	000024	001120			MOV	#20.,$ICNT	;20 ITERATIONS
024700	112737	000001	001131			MOVB	#1,$ERMAX	;ONE ERROR ALLOWED
024706	012737	024722	001122			MOV	#T41,$LPADR	;LOAD LOOP ON TEST ADDRESS
024714	012737	024722	001124			MOV	#T41,$LPERR	;LOAD LOOP ON ERROR ADDRESS
024722					T41:
024722	012706	001100				MOV	#STACK,SP	;LOAD THE STACK POINTER
024726	013700	001276				MOV	$BASE,R0	;R0 = UNIBUS ADDRESS OF UUT
024732	013701	001456				MOV	TSTQUE,R1	;R1 = POINTER TO DEVICE
024736	010037	001136				MOV	R0,$BDADR	;SETUP REGISTER ADDRESS
024742	062737	000012	001136			ADD	#RMDS,$BDADR
					;	CLEAR THE MASSBUS
					;	SET FORMAT
					;	LOAD LAST TRACK AND SECTOR
					;	LOAD LAST CYLINDER
					;
					;	VERIFY THAT "LBT" IS RESET
024750	012760	000040	000010			MOV	#CLR,RMCS2(R0)	;CLEAR THE MASSBUS
024756	111160	000010				MOVB	(R1),RMCS2(R0)	;SELECT UNIT

024762	012760	000000	000032			MOV	#0,RMOF(R0)	;LOAD RMOF

024770	012760	002035	000006			MOV	#002035,RMDA(R0)	;LOAD RMDA

```

```
     024776  012760  001466  000034            MOV     #822.,RMDC(R0)  ;LOAD RMDC

     025004  016037  000012  001142            MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
     025012  042737  175777  001142            BIC     #↑CLBT,$BDDAT
     025020  001403                            BEQ     10$             ;BRANCH IF LBT IS RESET
     025022  005037  001140                    CLR     $GDDAT          ;LBT SHOULD BE ZERO
     025026  104141                            ERROR   141             ;CANNOT RESET "LBT"

     025030                            10$:
                                       ;       ENABLE DEBUG CLOCK
                                       ;       SET GO
                                       ;       FORCE EBL
                                       ;
                                       ;       VERIFY THAT LBT IS SET

     025030  012760  000001  000024            MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

     025036  012760  000000  000014            MOV     #0,RMER1(R0)    ;LOAD RMER1

     025044  012760  000000  000042            MOV     #0,RMER2(R0)    ;LOAD RMER2

     025052  012760  000001  000000            MOV     #GO,RMCS1(R0)   ;LOAD RMCS1

     025060  012760  060001  000024            MOV     #DMD!DBEN!DEBL,RMMR1(R0)        ;LOAD RMMR1

     025066  012760  040001  000024            MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1

     025074  016037  000012  001142            MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
     025102  042737  175777  001142            BIC     #↑CLBT,$BDDAT
     025110  001004                            BNE     20$             ;BRANCH IF LBT IS SET
     025112  012737  002000  001140            MOV     #LBT,$GDDAT
     025120  104142                            ERROR   142             ;CANT SET LBT

     025122                            20$:                            ;END OF TEST


                                       ;;*****************************************************
                                       ;*TEST 42        COMPOSITE ERROR TEST

                                       ;;*****************************************************
     025122  000004                    TST42:  SCOPE
     025124  012737  000042  001226            MOV     #42,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

     025132  000240                            NOP
     025134  012737  000024  001120            MOV     #20.,$ICNT      ;20 ITERATIONS
     025142  112737  000001  001131            MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
     025150  012737  025164  001122            MOV     #T42,$LPADR   ;LOAD LOOP ON TEST ADDRESS
     025156  012737  025164  001124            MOV     #T42,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
     025164                            T42:
     025164  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
     025170  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
     025174  013701  001456                    MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
     025200  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
     025206  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
     025212  010037  001136                    MOV     R0,$BDADR       ;SETUP REGISTER ADDRESS
     025216  062737  000012  001136            ADD     #RMDS,$BDADR
                                       ;USING DIAGNOSTIC MODE, CLEAR ALL ERRORS AND VERIFY THAT COMPOSITE
```

```
                                                ;ERROR IS RESET.

025224  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

025232  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1

025240  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

025246  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
025254  042737  137777  001142                  BIC     #↑CERR,$BDDAT
025262  001403                                  BEQ     10$             ;BRANCH IF ERR IS RESET
025264  005037  001140                          CLR     $GDDAT
025270  104143                                  ERROR   143             ;CANT READ ZERO FROM ERR
025272  012737  040000  001140          10$:    MOV     #ERR,$GDDAT
                                        ;SET BOTH ERROR REGISTERS AND VERIFY THAT COMPOSITE ERROR IS SET

025300  012760  177777  000014                  MOV     #-1,RMER1(R0)   ;LOAD RMER1

025306  012760  177777  000042                  MOV     #-1,RMER2(R0)   ;LOAD RMER2

025314  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
025322  042737  137777  001142                  BIC     #↑CERR,$BDDAT
025330  001001                                  BNE     20$             ;BRANCH IF ERR IS SET
025332  104144                                  ERROR   144             ;CANT SET ERR
025334                                  20$:
                                        ;VERIFY THAT COMPOSITE ERROR SETS FOR EACH BIT OF RMER1
025334  012702  000001                          MOV     #1,R2           ;INITIALIZE TEST PATTERN
025340                                  30$:
                                        ;       WRITE THE TEST PATTERN AND VERIFY THAT ERR IS SET
025340  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
025346  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

025352  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

025360  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1

025366  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

025374  010260  000014                          MOV     R2,RMER1(R0)    ;LOAD RMER1

025400  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
025406  042737  137777  001142                  BIC     #↑CERR,$BDDAT
025414  001005                                  BNE     40$             ;BRANCH IF COMPOSITE ERROR SET
025416  010237  001174                          MOV     R2,$TMP0        ;SAVE RMER1 TEST PATTERN
025422  005037  001176                          CLR     $TMP1           ;SAVE RMER2 TEST PATTERN
025426  104145                                  ERROR   145             ;ERR NOT SET BY RMER1 ERROR
025430                                  40$:
                                        ;       ADVANCE THE TEST PATTERN FOR RMER1
025430  006302                                  ASL     R2
025432  001342                                  BNE     30$             ;CONTINUE IF TEST NOT DONE
025434                                  50$:
                                        ;VERIFY THAT COMPOSITE ERROR SETS FOR EACH BIT OF RMER2
025434  012702  000001                          MOV     #1,R2           ;INITIALIZE TEST PATTERN
025440                                  60$:
                                        ;       WRITE THE TEST PATTERN AND VERIFY THAT ERR IS SET
025440  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
```

```
025446  111160  000010                       MOVB    (R1),RMCS2(R0)   ;SELECT UNIT
025452  012760  000001  000024               MOV     #DMD,RMMR1(R0)   ;LOAD RMMR1
025460  012760  000000  000014               MOV     #0,RMER1(R0)     ;LOAD RMER1
025466  012760  000000  000042               MOV     #0,RMER2(R0)     ;LOAD RMER2
025474  010260  000042                       MOV     R2,RMER2(R0)     ;LOAD RMER2
025500  016037  000012  001142               MOV     RMDS(R0),$BDDAT  ;STORE RMDS AT $BDDAT
025506  042737  137777  001142               BIC     #↑CERR,$BDDAT
025514  012737  040000  001140               MOV     #ERR,$GDDAT      ;SETUP EXPECTED VALUE FOR COMP ERROR
025522  032702  001567                       BIT     #XNUER2,R2
025526  001402                               BEQ     65$              ;BRANCH IF TEST BIT IS A USED BIT
025530  005037  001140                       CLR     $GDDAT           ;TEST BIT IS NOT USED - ERR SHOULD BE 0
025534  023737  001140  001142       65$:    CMP     $GDDAT,$BDDAT
025542  001405                               BEQ     70$              ;BRANCH IF COMP ERROR IS OK
025544  005037  001174                       CLR     $TMP0            ;SAVE RMER1 TEST PATTERN
025550  010237  001176                       MOV     R2,$TMP1         ;SAVE RMER2 TEST PATTERN
025554  104145                               ERROR   145              ;ERR NOT SET BY RMER2 ERROR
025556                               70$:
                                     ;       ADVANCE THE TEST PATTERN FOR RMER2
025556  006302                               ASL     R2
025560  001327                               BNE     60$              ;CONTINUE IF TEST NOT DONE
025562                               80$:                             ;END OF TEST
                                     ;;*****************************************************************
                                     ;*TEST 43        WRITE GO TEST
                                     ;;*****************************************************************
025562  000004                       TST43:  SCOPE
025564  012737  000043  001226               MOV     #43,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
025572  000240                               NOP
025574  012737  000024  001120               MOV     #20.,$ICNT       ;20 ITERATIONS
025602  112737  000001  001131               MOVB    #1,$ERMAX        ;ONE ERROR ALLOWED
025610  012737  025624  001122               MOV     #T43,$LPADR      ;LOAD LOOP ON TEST ADDRESS
025616  012737  025624  001124               MOV     #T43,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
025624                               T43:
025624  012706  001100                       MOV     #STACK,SP        ;LOAD THE STACK POINTER
025630  013700  001276                       MOV     $BASE,R0         ;R0 = UNIBUS ADDRESS OF UUT
025634  013701  001456                       MOV     TSTQUE,R1        ;R1 = POINTER TO DEVICE
025640  010037  001136                       MOV     R0,$BDADR        ;COPY RMCS1 ADDRESS
025644  005002                               CLR     R2               ;INITIALIZE FUNCTION CODE
025646                               10$:
                                     ;CLEAR THE MASSBUS, SET DIAGNOSTIC MODE AND ENABLE DEBUG CLOCK
025646  012760  000040  000010               MOV     #CLR,RMCS2(R0)   ;CLEAR THE MASSBUS
025654  111160  000010                       MOVB    (R1),RMCS2(R0)   ;SELECT UNIT
025660  012760  000001  000024               MOV     #DMD,RMMR1(R0)   ;LOAD RMMR1
025666  012760  041001  000024               MOV     #DMD!DBEN!MWR,RMMR1(R0) ;LOAD RMMR1
```

```
025674  012760  000000  000014          MOV     #0,RMER1(R0)      ;LOAD RMER1

025702  012760  000000  000042          MOV     #0,RMER2(R0)      ;LOAD RMER2
                                ;TRANSFER THE FUNCTION CODE AND GO BIT TO RMCS1, VERIFY GO IS SET
025710  010203                          MOV     R2,R3             ;SETUP FUNCTION CODE
025712  052703  000001                  BIS     #GO,R3

025716  010360  000000                  MOV     R3,RMCS1(R0)      ;LOAD RMCS1

025722  016037  000000  001142          MOV     RMCS1(R0),$BDDAT          ;STORE RMCS1 AT $BDDAT
025730  032737  000001  001142          BIT     #GO,$BDDAT
025736  001007                          BNE     20$               ;BRANCH IF GO IS SET
                                ;REPORT THE ERROR-CANT SET GO WITH THIS FUNCTION CODE
025740  042737  177700  001142          BIC     #^CFNCMSK,$BDDAT
025746  010337  001140                  MOV     R3,$GDDAT         ;SAVE FUNCTION CODE
025752  104146                          ERROR   146               ;CANT SET GO
025754  000405                          BR      30$
025756                          20$:
                                ;ADVANCE R2 TO THE NEXT FUNCTION CODE
025756  062702  000002                  ADD     #2,R2
025762  022702  000076                  CMP     #ILF76,R2
025766  103327                          BHIS    10$

025770                          30$:                              ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 44       BRANCH MULTIPLEXOR TEST

                                ;;*****************************************************************
025770  000004                  TST44:  SCOPE
025772  012737  000044  001226          MOV     #44,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX

026000  000240                          NOP
026002  012737  000024  001120          MOV     #20.,$ICNT        ;20 ITERATIONS
026010  112737  000001  001131          MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
026016  012737  026032  001122          MOV     #T44,$LPADR  ;LOAD LOOP ON TEST ADDRESS
026024  012737  026032  001124          MOV     #T44,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
026032                          T44:
026032  012706  001100                  MOV     #STACK,SP         ;LOAD THE STACK POINTER
026036  013700  001276                  MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
026042  013701  001456                  MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
026046  010037  001136                  MOV     R0,$BDADR         ;COPY REGISTER ADDRESS
026052  062737  000040  001136          ADD     #RMMR2,$BDADR
026060  012702  026312                  MOV     #100$,R2          ;INITIALIZE TABLE POINTER
                                ;CLEAR THE MASSBUS AND SET DEBUG CLOCK ENABLE
026064                          10$:
026064  012760  000040  000010          MOV     #CLR,RMCS2(R0)    ;CLEAR THE MASSBUS
026072  111160  000010                  MOVB    (R1),RMCS2(R0)    ;SELECT UNIT

026076  012760  000001  000024          MOV     #DMD,RMMR1(R0)    ;LOAD RMMR1

026104  012760  041001  000024          MOV     #DMD!DBEN!MUR,RMMR1(R0) ;LOAD RMMR1

026112  012760  000000  000014          MOV     #0,RMER1(R0)      ;LOAD RMER1

026120  012760  000000  000042          MOV     #0,RMER2(R0)      ;LOAD RMER2
```

```
                                        ;THE TEST BIT SHOULD BE ONE BECAUSE THE ADDRESS IS ALL ONES WHEN
                                        ;THE COMMAND SEQUENCER IS INITIALIZED.

026126  016037  000040  001142                  MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
026134  032737  010000  001142                  BIT     #TST,$BDDAT
026142  001010                                  BNE     15$                     ;BRANCH IF TEST BIT IS ON
026144  042737  167777  001142                  BIC     #↑CTST,$BDDAT           ;SETUP FOR ERROR TYPE
026152  012737  010000  001140                  MOV     #TST,$GDDAT
026160  104147                                  ERROR   147                     ;TEST BIT NOT SET
026162  000452                                  BR      40$                     ;SKIP REST OF TEST
026164                                  15$:
                                        ;GET THE FUNCTION CODE FROM THE TABLE AND TRANSFER IT TO THE DEVICE,
                                        ;THEN STEP THE COMMAND SEQUENCER ACCORDING TO THE TABLE.

026164  111203                                  MOVB    (R2),R3
026166  052703  000001                          BIS     #GO,R3
026172  042703  177700                          BIC     #↑CFNCMSK,R3            ;R3=FUNCTION CODE, GO BIT

026176  010360  000000                          MOV     R3,RMCS1(R0)            ;LOAD RMCS1
026202  010337  001174                          MOV     R3,$TMP0                ;SAVE R3 FOR ERROR MSG

026206  116203  000001                          MOVB    1(R2),R3                ;GET CLOCK COUNT IN R3
026212  042703  177400                          BIC     #↑C377,R3
026216                                  20$:

026216  012760  141001  000024                  MOV     #DMD!DBEN!MUR!DBCK,RMMR1(R0)    ;LOAD RMMR1

026224  012760  041001  000024                  MOV     #DMD!DBEN!MUR,RMMR1(R0) ;LOAD RMMR1
026232  005303                                  DEC     R3                      ;DECREMENT CLOCK COUNT
026234  001370                                  BNE     20$                     ;ISSUE CLOCKS TILL ZERO

                                        ;GET THE TEST BIT AND COMPARE IT WITH THE TABLE ENTRY

026236  016037  000040  001142                  MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
026244  042737  167777  001142                  BIC     #↑CTST,$BDDAT
026252  016237  000002  001140                  MOV     2(R2),$GDDAT
026260  023737  001140  001142                  CMP     $GDDAT,$BDDAT
026266  001402                                  BEQ     30$                     ;BRANCH IF TEST BIT OK
026270  104150                                  ERROR   150                     ;TEST BIT IS INCORRECT
026272  000406                                  BR      40$                     ;SKIP REST OF TEST
026274                                  30$:
                                        ;MOVE THE TABLE POINTER AND CONTINUE IF NEXT ENTRY POSITIVE
026274  062702  000004                          ADD     #4,R2
026300  105762  000001                          TSTB    1(R2)
026304  100401                                  BMI     40$                     ;BRANCH IF DONE TEST
026306  000666                                  BR      10$                     ;REPEAT TEST
026310  000436                          40$:    BR      200$                    ;JUMP OVER TABLE

026312                                  100$:

                                        ;TABLE OF FUNCTION CODES, CLOCK COUNTS, AND TEST BITS

026312     000                                  .BYTE   NOP                     ;MUX ADDRESS=DATA COMMAND
026313     001                                  .BYTE   1
026314  000000                                  .WORD   0                       ;TEST BIT=0
```

```
026316      000             .BYTE   NOP             ;MUX ADDRESS=UNIT READY
026317      002             .BYTE   2
026320   000000             .WORD   0               ;TEST BIT=0

026322      010             .BYTE   DRVCLR          ;MUX ADDRESS=F4
026323      001             .BYTE   1
026324   010000             .WORD   TST             ;TEST BIT=1

026326      050             .BYTE   WCD             ;MUX ADDRESS=F4
026327      001             .BYTE   1
026330   000000             .WORD   0               ;TEST BIT=0

026332      012             .BYTE   RLEASE          ;MUX ADDRESS=F4
026333      001             .BYTE   1
026334   010000             .WORD   TST             ;TEST BIT=1

026336      052             .BYTE   WCH             ;MUX ADDRESS=F4
026337      001             .BYTE   1
026340   000000             .WORD   0               ;TEST BIT=0

026342      020             .BYTE   RIP             ;MUX ADDRESS=F4
026343      001             .BYTE   1
026344   010000             .WORD   TST             ;TEST BIT=1

026346      060             .BYTE   WD              ;MUX ADDRESS=F4
026347      001             .BYTE   1
026350   000000             .WORD   0               ;TEST BIT=0

026352      022             .BYTE   PAKACK          ;MUX ADDRESS=F4
026353      001             .BYTE   1
026354   010000             .WORD   TST             ;TEST BIT=1

026356      062             .BYTE   WH              ;MUX ADDRESS=F4
026357      001             .BYTE   1
026360   000000             .WORD   0               ;TEST BIT=0

026362      030             .BYTE   SEARCH          ;MUX ADDRESS=F4
026363      001             .BYTE   1
026364   010000             .WORD   TST             ;TEST BIT=1

026366      070             .BYTE   RD              ;MUX ADDRESS=F4
026367      001             .BYTE   1
026370   000000             .WORD   0               ;TEST BIT=0

026372      032             .BYTE   ILF32           ;MUX ADDRESS=F4
026373      001             .BYTE   1
026374   010000             .WORD   TST             ;TEST BIT=1

026376      072             .BYTE   RH              ;MUX ADDRESS=F4
026377      001             .BYTE   1
026400   000000             .WORD   0               ;TEST BIT=0

026402      000             .BYTE                   ;END OF TABLE
026403      377             .BYTE   #-1
026404   000000             .WORD
026406                200$:                         ;END OF TEST
```

```


                                       ;;*******************************************************
                                       ;*TEST 45          SET/RESET GO TEST

                                       ;;*******************************************************
026406  000004                         TST45:  SCOPE
026410  012737  000045  001226                 MOV     #45,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX

026416  000240                                 NOP
026420  012737  000024  001120                 MOV     #20.,$ICNT        ;20 ITERATIONS
026426  112737  000001  001131                 MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
026434  012737  026450  001122                 MOV     #T45,$LPADR  ;LOAD LOOP ON TEST ADDRESS
026442  012737  026450  001124                 MOV     #T45,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
026450                                 T45:
026450  012706  001100                         MOV     #STACK,SP         ;LOAD THE STACK POINTER
026454  013700  001276                         MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
026460  013701  001456                         MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
026464  012702  027106                         MOV     #200$,R2          ;INITIALIZE FUNCTION CODE POINTER
                                       ;CLEAR, THEN SET DIAGNOSTIC MODE, CLEAR COMPOSITE ERROR, SET MEDIUM
                                       ;ON LINE AND ENABLE DEBUG CLOCK
026470                                 10$:
026470  012760  000040  000010                 MOV     #CLR,RMCS2(R0)    ;CLEAR THE MASSBUS
026476  111160  000010                         MOVB    (R1),RMCS2(R0)    ;SELECT UNIT

026502  012760  000001  000024                 MOV     #DMD,RMMR1(R0)    ;LOAD RMMR1

026510  012760  041001  000024                 MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

026516  012760  000000  000014                 MOV     #0,RMER1(R0)      ;LOAD RMER1

026524  012760  000000  000042                 MOV     #0,RMER2(R0)      ;LOAD RMER2
                                       ;TRANSFER THE FUNCTION CODE AND GO BIT TO RMCS1 AND VERIFY GO IS SET
026532  111203                                 MOVB    (R2),R3           ;GET FUNCTION CODE
026534  042703  177701                         BIC     #^CILF76,R3       ;CLEAR UNUSED BITS
026540  052703  000001                         BIS     #GO,R3            ;SET GO

026544  010360  000000                         MOV     R3,RMCS1(R0)      ;LOAD RMCS1

026550  016037  000000  001142                 MOV     RMCS1(R0),$BDDAT          ;STORE RMCS1 AT $BDDAT
026556  032737  000001  001142                 BIT     #GO,$BDDAT
026564  001011                                 BNE     20$               ;BRANCH IF GO IS SET
026566  042737  177700  001142                 BIC     #^CFNCMSK,$BDDAT
026574  010337  001140                         MOV     R3,$GDDAT         ;SAVE EXPECTED RESULT
026600  010037  001136                         MOV     R0,$BDADR         ;COPY REGISTER ADDRESS
026604  104151                                 ERROR   151               ;CANT SET GO
026606  000536                                 BR      100$

026610                                 20$:
                                       ;GET READY STATUS AND VERIFY THAT IT IS THE COMPLEMENT OF GO
026610  005037  001140                         CLR     $GDDAT            ;EXPECT DRY TO BE OFF
026614  032737  000001  001142                 BIT     #GO,$BDDAT        ;WAS GO SET??
026622  001003                                 BNE     30$               ;YES!!
026624  012737  000200  001140                 MOV     #DRY,$GDDAT       ;GO WAS NOT SET, DRY SHOULD BE
026632                                 30$:

026632  016037  000012  001142                 MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
```

```
026640 042737 177577 001142             BIC     #↑CDRY,$BDDAT
026646 023737 001140 001142             CMP     $GDDAT,$BDDAT
026654 001406                           BEQ     40$                     ;BRANCH IF DRY IS OK
026656 010037 001136                    MOV     R0,$BDADR               ;COPY REGISTER ADDRESS
026662 062737 000012 001136             ADD     #RMDS,$BDADR
026670 104152                           ERROR   152                     ;DRY NOT COMPLEMENT OF GO
026672                          40$:
                                ;STEP THE DEBUG CLOCK AND VERIFY THAT GO REMAINS SET
026672 116204 000001                    MOVB    1(R2),R4                ;GET NUMBER OF CLOCK CYCLES
026676 042704 177400                    BIC     #↑C377,R4
026702                          50$:

026702 012760 141001 000024             MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

026710 012760 041001 000024             MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

026716 016037 000000 001142             MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
026724 042737 177700 001142             BIC     #↑CFNCMSK,$BDDAT        ;CLEAR UNUSED BITS
026732 010037 001136                    MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS
026736 010337 001140                    MOV     R3,$GDDAT               ;SAVE EXPECTED RESULT
                                ;DECREMENT CLOCK COUNT AND EXIT LOOP IF ZERO
026742 005304                           DEC     R4
026744 001406                           BEQ     60$
026746 032737 000001 001142             BIT     #GO,$BDDAT              ;IS GO STILL SET??
026754 001352                           BNE     50$                     ;YES!!
026756 104153                           ERROR   153                     ;GO RESET EARLY
026760 000451                           BR      100$                    ;OUT OF SYNC=SKIP TO NEXT
026762                          60$:
                                ;GO SHOULD NOW BE RESET AND DRY SHOULD BE SET
026762 032737 000001 001142             BIT     #GO,$BDDAT              ;IS GO RESET??
026770 001405                           BEQ     70$                     ;YES!!
026772 042737 000001 001140             BIC     #GO,$GDDAT              ;SETUP EXPECTED RESULT
027000 104154                           ERROR   154                     ;GO DID NOT RESET
027002 000440                           BR      100$
027004                          70$:
027004 012737 000200 001140             MOV     #DRY,$GDDAT             ;EXPECT DRIVE READY TO BE SET
027012 032737 000001 001142             BIT     #GO,$BDDAT              ;DID GO RESET??
027020 001402                           BEQ     80$                     ;YES!!
027022 005037 001140                    CLR     $GDDAT                  ;GO IS SET-
027026                          80$:

027026 016037 000012 001142             MOV     RMDS(R0),$BDDAT         ;STORE RMDS AT $BDDAT
027034 042737 177577 001142             BIC     #↑CDRY,$BDDAT
027042 010037 001136                    MOV     R0,$BDADR               ;COPY REGISTER ADDRESS
027046 062737 000012 001136             ADD     #RMDS,$BDADR
027054 023737 001140 001142             CMP     $GDDAT,$BDDAT           ;IS DRIVE READY OK??
027062 001401                           BEQ     90$                     ;YES!!
027064 104152                           ERROR   152                     ;DRY NOT COMPLEMENT OF GO
027066                          90$:
                                ;ADVANCE TO THE NEXT FUNCTION CODE TO BE TESTED-EXIT IF DONE
027066 062702 000002                    ADD     #2,R2                   ;MOVE TABLE POINTER
027072 105762 000001                    TSTB    1(R2)                   ;END OF TABLE??
027076 100402                           BMI     100$                    ;YES!!
027100 000137 026470                    JMP     10$                     ;TEST THIS FUNCTION CODE
027104 000423                   100$:   BR      300$                    ;GO TO NEXT TEST
```

```
027106                  200$:
                        ;;*************************************************************
                        ;TABLE OF FUNCTION CODES AND CLOCK COUNTS USED DURING TEST
027106      002                 .BYTE   ILF02           ;ILLEGAL FUNCTION CODE #2
027107      001                 .BYTE   1

027110      004                 .BYTE   SEEK            ;SEEK COMMAND
027111      001                 .BYTE   1

027112      006                 .BYTE   RECAL           ;RECALIBRATE COMMAND
027113      001                 .BYTE   1

027114      014                 .BYTE   OFFSET          ;OFFSET COMMAND
027115      001                 .BYTE   1

027116      016                 .BYTE   RTC             ;RETURN TO CENTER LINE COMMAND
027117      001                 .BYTE   1

027120      024                 .BYTE   ILF24           ;ILLEGAL FUNCTION CODE #24
027121      001                 .BYTE   1

027122      026                 .BYTE   ILF26           ;ILLEGAL FUNCTION CODE #26
027123      001                 .BYTE   1

027124      034                 .BYTE   ILF34           ;ILLEGAL FUNCTION CODE #34
027125      001                 .BYTE   1

027126      036                 .BYTE   ILF36           ;ILLEGAL FUNCTION CODE #36
027127      001                 .BYTE   1

027130      042                 .BYTE   ILF42           ;ILLEGAL FUNCTION CODE #42
027131      001                 .BYTE   1

027132      044                 .BYTE   ILF44           ;ILLEGAL FUNCTION CODE #44
027133      001                 .BYTE   1

027134      046                 .BYTE   ILF46           ;ILLEGAL FUNCTION CODE #46
027135      001                 .BYTE   1

027136      054                 .BYTE   ILF54           ;ILLEGAL FUNCTION CODE #54
027137      001                 .BYTE   1

027140      056                 .BYTE   ILF56           ;ILLEGAL FUNCTION CODE #56
027141      001                 .BYTE   1

027142      064                 .BYTE   ILF64           ;ILLEGAL FUNCTION CODE #64
027143      001                 .BYTE   1

027144      066                 .BYTE   ILF66           ;ILLEGAL FUNCTION CODE #66
027145      001                 .BYTE   1

027146      074                 .BYTE   ILF74           ;ILLEGAL FUNCTION CODE #74
027147      001                 .BYTE   1

027150      076                 .BYTE   ILF76           ;ILLEGAL FUNCTION CODE #76
027151      001                 .BYTE   1
```

```
027152     000                          .BYTE                      ;END OF TABLE
027153     377                          .BYTE   #-1

027154                          300$:                              ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 46        END 1 RESET GO TEST

                                ;;*****************************************************************
027154  000004                  TST46:  SCOPE
027156  012737  000046  001226          MOV     #46,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX

027164  000240                          NOP
027166  012737  000024  001120          MOV     #20.,$ICNT         ;20 ITERATIONS
027174  112737  000001  001131          MOVB    #1,$ERMAX          ;ONE ERROR ALLOWED
027202  012737  027216  001122          MOV     #T46,$LPADR    ;LOAD LOOP ON TEST ADDRESS
027210  012737  027216  001124          MOV     #T46,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
027216                          T46:
027216  012706  001100                  MOV     #STACK,SP          ;LOAD THE STACK POINTER
027222  013700  001276                  MOV     $BASE,R0           ;R0 = UNIBUS ADDRESS OF UUT
027226  013701  001456                  MOV     TSTQUE,R1          ;R1 = POINTER TO DEVICE
027232  012702  027476                  MOV     #100$,R2           ;INITIALIZE TABLE POINTER
027236  010037  001136                  MOV     R0,$BDADR          ;COPY RMCS1 ADDRESS

027242                          10$:
                                ;CLEAR MASSBUS, THEN SET MEDIUM ON LINE AND ENABLE DEBUG CLOCK
027242  012760  000040  000010          MOV     #CLR,RMCS2(R0)     ;CLEAR THE MASSBUS
027250  111160  000010                  MOVB    (R1),RMCS2(R0)     ;SELECT UNIT

027254  012760  000001  000024          MOV     #DMD,RMMR1(R0)     ;LOAD RMMR1

027262  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

027270  012760  000000  000014          MOV     #0,RMER1(R0)       ;LOAD RMER1

027276  012760  000000  000042          MOV     #0,RMER2(R0)       ;LOAD RMER2
                                ;TRANSFER THE FUNCTION CODE AND GO BIT TO RMCS1, VERIFY GO IS SET
027304  111203                          MOVB    (R2),R3            ;GET FUNCTION CODE FROM
027306  042703  177701                  BIC     #↑C1LF76,R3        ;TABLE AND SET GO
027312  052703  000001                  BIS     #GO,R3
027316  010337  001140                  MOV     R3,$GDDAT          ;SAVE FUNCTION CODE FOR MSG

027322  010360  000000                  MOV     R3,RMCS1(R0)       ;LOAD RMCS1

027326  016037  000000  001142          MOV     RMCS1(R0),$BDDAT           ;STORE RMCS1 AT $BDDAT
027334  032737  000001  001142          BIT     #GO,$BDDAT
027342  001005                          BNE     20$                ;BRANCH IF GO IS SET
027344  042737  177700  001142          BIC     #↑CFNCMSK,$BDDAT
027352  104151                          ERROR   151                ;GO DID NOT SET
027354  000447                          BR      60$                ;OUT OF SYNC-SKIP
027356                          20$:
                                ;GET THE NUMBER OF CLOCK CYCLES FROM THE TABLE, SAVE EXPECTED STATUS
027356  116204  000001                  MOVB    1(R2),R4           ;R4=CLOCK COUNT
027362  042704  177400                  BIC     #↑C377,R4
027366                          30$:
```

```
                                        ;STEP THE DEBUG CLOCK AND VERIFY GO STATUS ON UNTIL CLOCK COUNT EXPIRES.

027366  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

027374  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

027402  016037  000000  001142                  MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
027410  042737  177700  001142                  BIC     #↑CFNCMSK,$BDDAT
027416  005304                                  DEC     R4
027420  001406                                  BEQ     40$             ;BRANCH IF GO SHOULD BE OFF
027422  032737  000001  001142                  BIT     #GO,$BDDAT
027430  001356                                  BNE     30$             ;CONTINUE IF GO IS ON
027432  104153                                  ERROR   153             ;GO RESET EARLY
027434  000417                                  BR      60$             ;OUT OF SYNC-SKIP
027436                                  40$:
                                        ;VERIFY THAT GO RESET AT END1
027436  032737  000001  001142                  BIT     #GO,$BDDAT      ;DID GO RESET??
027444  001405                                  BEQ     50$             ;YES!!
027446  042737  000001  001140                  BIC     #GO,$GDDAT
027454  104154                                  ERROR   154             ;GO NOT RESET AT END1
027456  000406                                  BR      60$
027460                                  50$:
                                        ;GET THE NEXT FUNCTION CODE FROM THE TABLE
027460  062702  000002                          ADD     #2,R2
027464  105762  000001                          TSTB    1(R2)
027470  100401                                  BMI     60$             ;BRANCH IF END OF TABLE
027472  000663                                  BR      10$             ;TEST THIS FUNCTION CODE
027474  000404                          60$:    BR      200$            ;JUMP OVER TABLE

027476                                  100$:
                                        ;TABLE OF FUNCTION CODES AND CLOCK COUNTS USED DURING TEST

027476     012                                  .BYTE   RLEASE          ;RELEASE COMMAND
027477     002                                  .BYTE   2

027500     030                                  .BYTE   SEARCH          ;SEARCH COMMAND
027501     002                                  .BYTE   2

027502     032                                  .BYTE   ILF32           ;ILLEGAL FUNCTON #32
027503     002                                  .BYTE   2

027504     000                                  .BYTE                   ;END OF TABLE
027505     377                                  .BYTE   #-1
027506                                  200$:                           ;END OF TEST


                                        ;;*****************************************************************
                                        ;*TEST 47       SET PULSE TEST

                                        ;;*****************************************************************
027506  000004                          TST47:  SCOPE
027510  012737  000047  001226                  MOV     #47,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

027516  000240                                  NOP
027520  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
027526  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
027534  012737  027550  001122                  MOV     #T47,$LPADR  ;LOAD LOOP ON TEST ADDRESS
```

```
027542  012737  027550  001124          MOV     #T47,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
027550                          T47:
027550  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
027554  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
027560  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
027564  010037  001136                  MOV     R0,$BDADR       ;COPY REG ADDRESS FOR MSG
027570  062737  000024  001136          ADD     #RMMR1,$BDADR
027576  012702  030042                  MOV     #100$,R2        ;INITIALIZE TABLE POINTER
027602                          10$:
                                ;CLEAR THE MASS BUS, ENABLE DEBUG CLOCK, AND RESET ERROR REGISTERS
027602  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
027610  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

027614  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

027622  012760  041001  000024          MOV     #DMD!DBEN!MUR,RMMR1(R0) ;LOAD RMMR1

027630  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

027636  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
                                ;VERIFY THAT CONTINUE, "CONT" IS RESET AFTER CLEAR

027644  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
027652  042737  177677  001142          BIC     #^CCONT,$BDDAT
027660  001404                          BEQ     20$             ;BRANCH IF CONT WAS CLEARED
027662  005037  001140                  CLR     $GDDAT          ;FOR ERROR MSG
027666  104155                          ERROR   155             ;CANT CLEAR CONTINUE
027670  000463                          BR      70$
027672                          20$:
                                ;GET THE FUNCTION CODE FROM THE TABLE AND TRANSFER IT TO RMCS1
027672  111203                          MOVB    (R2),R3
027674  052703  000001                  BIS     #GO,R3
027700  042703  177700                  BIC     #^CFNCMSK,R3    ;R3=FUNCTION CODE AND GO

027704  010360  000000                  MOV     R3,RMCS1(R0)    ;LOAD RMCS1
027710  010337  001174                  MOV     R3,$TMP0        ;SAVE FUNCTION CODE FOR MSG
                                ;GET THE CLOCK COUNT FROM THE TABLE
027714  116203  000001                  MOVB    1(R2),R3
027720  042703  177400                  BIC     #^C377,R3
                                ;GET THE BIT STREAM FOR CONTINUE FROM THE TABLE
027724  016204  000002                  MOV     2(R2),R4
027730                          30$:
                                ;STEP THE COMMAND SEQUENCER AND VERIFY CONTINUE STATUS

027730  012760  141001  000024          MOV     #DMD!DBEN!MUR!DBCK,RMMR1(R0)    ;LOAD RMMR1

027736  012760  041001  000024          MOV     #DMD!DBEN!MUR,RMMR1(R0) ;LOAD RMMR1

027744  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
027752  042737  177677  001142          BIC     #^CCONT,$BDDAT
027760  005037  001140                  CLR     $GDDAT          ;GENERATE EXPECTED CONTINUE
027764  032704  000001                  BIT     #BIT0,R4
027770  001403                          BEQ     40$
027772  012737  000100  001140          MOV     #CONT,$GDDAT
030000  023737  001140  001142  40$:    CMP     $GDDAT,$BDDAT
030006  001402                          BEQ     50$             ;BRANCH IF CONTINUE IS OK
```

```
030010  104156                  ERROR   156             ;CONTINUE IS INCORRECT
030012  000412                  BR      70$             ;SKIP
030014                  50$:
                        ;DECREMENT CLOCK COUNT AND SHIFT BIT STREAM
030014  005303                  DEC     R3
030016  001402                  BEQ     60$             ;BRANCH IF CLOCK COUNT EXPIRED
030020  006204                  ASR     R4              ;SHIFT TO NEXT CONTINUE BIT
030022  000742                  BR      30$             ;TEST NEXT CLOCK CYCLE
030024                  60$:
                        ;ADVANCE TABLE POINTER-EXIT IF DONE
030024  062702  000004          ADD     #4,R2
030030  105762  000001          TSTB    1(R2)
030034  100401                  BMI     70$             ;EXIT IF CLOCK COUNT NEGATIVE
030036  000661                  BR      10$             ;CONTINUE TEST
030040  000442          70$:    BR      200$            ;JUMP OVER TABLE
030042                  100$:
                        ;TABLE OF FUNCTION CODES, CLOCK COUNTS AND CONTINUE BITS FOR TEST
030042     000                  .BYTE   NOP             ;NOP COMMAND
030043     004                  .BYTE   4               ;4 CLOCKS
030044  000000                  .WORD   ↑B0000          ;CONTINUE=0000
030046     002                  .BYTE   ILF02           ;ILLEGAL FUNCTION 2
030047     002                  .BYTE   2
030050  000000                  .WORD   ↑B00
030052     004                  .BYTE   SEEK            ;SEEK COMMAND
030053     002                  .BYTE   2
030054  000000                  .WORD   ↑B00
030056     006                  .BYTE   RECAL           ;RECALIBRATE COMMAND
030057     002                  .BYTE   2
030060  000000                  .WORD   ↑B00
030062     010                  .BYTE   DRVCLR          ;DRIVE CLEAR COMMAND
030063     002                  .BYTE   2
030064  000001                  .WORD   ↑B01
030066     012                  .BYTE   RLEASE          ;RELEASE COMMAND
030067     003                  .BYTE   3
030070  000000                  .WORD   ↑B000
030072     014                  .BYTE   OFFSET          ;OFFSET COMMAND
030073     002                  .BYTE   2
030074  000000                  .WORD   ↑B00
030076     016                  .BYTE   RTC             ;RETURN TO CENTER COMMAND
030077     002                  .BYTE   2
030100  000000                  .WORD   ↑B00
030102     020                  .BYTE   RIP             ;READ IN PRESET COMMAND
030103     004                  .BYTE   4
030104  000016                  .WORD   ↑B1110
030106     022                  .BYTE   PAKACK          ;PACK ACKNOWLEDGE
```

```
030107     004                          .BYTE   4
030110  000016                          .WORD   ↑B1110

030112     024                          .BYTE   ILF24           ;ILLEGAL FUNCTION 24
030113     002                          .BYTE   2
030114  000000                          .WORD   ↑B00

030116     026                          .BYTE   ILF26           ;ILLEGAL FUNCTION 26
030117     002                          .BYTE   2
030120  000000                          .WORD   ↑B00

030122     030                          .BYTE   SEARCH          ;SEARCH COMMAND
030123     003                          .BYTE   3
030124  000000                          .WORD   ↑B000

030126     032                          .BYTE   ILF32           ;ILLEGAL FUNCTION 32
030127     003                          .BYTE   3
030130  000000                          .WORD   ↑B000

030132     034                          .BYTE   ILF34           ;ILLEGAL FUNCTION 34
030133     002                          .BYTE   2
030134  000000                          .WORD   ↑B00

030136     036                          .BYTE   ILF36           ;ILLEGAL FUNCTION 36
030137     002                          .BYTE   2
030140  000000                          .WORD   ↑B00

030142     000                          .BYTE                   ;END OF TABLE
030143     377                          .BYTE   -1
030144  000000                          .WORD

030146                          200$:                           ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 50        SET/RESET IVC TEST

                                ;;*****************************************************************
030146  000004                  TST50:  SCOPE
030150  012737  000050  001226          MOV     #50,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

030156  000240                          NOP
030160  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
030166  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
030174  012737  030210  001122          MOV     #T50,$LPADR  ;LOAD LOOP ON TEST ADDRESS
030202  012737  030210  001124          MOV     #T50,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
030210                          T50:
030210  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
030214  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
030220  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
030224  010037  001136                  MOV     R0,$BDADR       ;SETUP REG ADDRESS
030230  062737  000042  001136          ADD     #RMER2,$BDADR
030236  005002                          CLR     R2              ;R2=FUNCTION CODE
030240                          10$:
                                ;INITIALIZE AND VERIFY THAT IVC STATUS IS ZERO.
030240  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
030246  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
```

```
030252  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
030260  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
030266  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
030274  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
030302  016037  000042  001142          MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
030310  042737  167777  001142          BIC     #↑CIVC,$BDDAT
030316  001404                          BEQ     20$             ;BRANCH IF IVC IS ZERO
030320  005037  001140                  CLR     $GDDAT
030324  104157                          ERROR   157             ;CANT CLEAR IVC
030326  000444                          BR      40$             ;SKIP REST OF TEST
030330                          20$:
                                ;LOAD THE FUNCTION CODE WITH GO BIT, STEP THE COMMAND SEQUENCER OFF
                                ;ADDRESS 0 AND VERIFY IVC STATUS.
030330  010203                          MOV     R2,R3           ;SETUP FUNCTION CODE
030332  052703  000001                  BIS     #GO,R3
030336  010360  000000                  MOV     R3,RMCS1(R0)    ;LOAD RMCS1
030342  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
030350  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
030356  016037  000042  001142          MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
030364  042737  167777  001142          BIC     #↑CIVC,$BDDAT   ;SET ACTUAL STATUS
030372  016237  065764  001140          MOV     FNCDTB(R2),$GDDAT  ;SETUP EXPECTED STATUS FROM
030400  042737  167777  001140          BIC     #↑CIVC,$GDDAT   ;FUNCTION CODE TABLE
030406  023737  001140  001142          CMP     $GDDAT,$BDDAT
030414  001403                          BEQ     30$             ;BRANCH IF IVC IS OK
030416  010237  001174                  MOV     R2,$TMP0        ;SAVE FUNCTION CODE FOR MSG
030422  104160                          ERROR   160             ;IVC IS INCORRECT
030424                          30$:
                                ;ADVANCE FUNCTION CODE AND REPEAT TEST IF NOT DONE
030424  062702  000002                  ADD     #2,R2
030430  022702  000076                  CMP     #ILF76,R2
030434  103401                          BLO     40$             ;BRANCH IF DONE TEST
030436  000700                          BR      10$

030440                          40$:                            ;END OF TEST

                                ;;*****************************************************
                                ;*TEST 51        SET LSC TEST

                                ;;*****************************************************
030440  000004                  TST51:  SCOPE
030442  012737  000051  001226          MOV     #51,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
030450  000240                          NOP
030452  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
030460  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
030466  012737  030502  001122          MOV     #T51,$LPADR  ;LOAD LOOP ON TEST ADDRESS
030474  012737  030502  001124          MOV     #T51,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
```

```
030502                                  T51:
030502  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
030506  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
030512  013701  001456                          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
030516  010037  001136                          MOV     R0,$BDADR
030522  062737  000042  001136                  ADD     #RMER2,$BDADR

                                        ;INITIALIZE AND VERIFY THAT LOSS OF SYSTEM CLOCK, "LSC", IS RESET
030530  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
030536  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

030542  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

030550  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1

030556  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1

030564  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

030572  016037  000042  001142                  MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
030600  042737  173777  001142                  BIC     #^CLSC,$BDDAT
030606  001403                                  BEQ     10$             ;BRANCH IF LSC IS ZERO
030610  005037  001140                          CLR     $GDDAT
030614  104161                                  ERROR   161             ;CANT CLEAR LSC
                                        ;WITH DEBUG CLOCK ENABLED, SET GO AND WAIT FOR ONE SHOT TO SET
030616                                  10$:

030616  012760  000001  000000                  MOV     #GO,RMCS1(R0)   ;LOAD RMCS1
030624  012737  000001  001524                  MOV     #1,WATCH        ;SET WATCHDOG TIMER VALUE
030632  004777  150670                          JSR     PC,@CLOCK       ;START THE CLOCK
030636  005737  001524                  20$:    TST     WATCH
030642  001375                                  BNE     20$             ;WAIT FOR WATCH ZERO
030644  004777  150660                          JSR     PC,@STOP        ;STOP THE CLOCK
                                        ;ONE SHOT SHOULD BE SET-DISABLE DIAGNOSTIC CLOCK AND LSC SHOULD SET.

030650  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

030656  016037  000042  001142                  MOV     RMER2(R0),$BDDAT        ;STORE RMER2 AT $BDDAT
030664  042737  173777  001142                  BIC     #^CLSC,$BDDAT
030672  001004                                  BNE     30$             ;BRANCH IF LSC SET
030674  012737  004000  001140                  MOV     #LSC,$GDDAT
030702  104162                                  ERROR   162             ;CANT SET LSC

030704                                  30$:                            ;END OF TEST

                                        ;;*****************************************************************
                                        ;*TEST 52       DECODE TEST

                                        ;;*****************************************************************
030704  000004                          TST52:  SCOPE
030706  012737  000052  001226                  MOV     #52,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

030714  000240                                  NOP
030716  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
030724  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
030732  012737  030746  001122                  MOV     #T52,$LPADR ;LOAD LOOP ON TEST ADDRESS
```

```
030740  012737  030746  001124          MOV     #T52,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
030746                          T52:
030746  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
030752  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
030756  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
030762  005037  001416                  CLR     RMER10          ;NO ERROR FIRST TEST
030766                          5$:
030766  004737  031346                  JSR     PC,100$         ;INITIALIZE

                                ;EXECUTE A PACK ACKNOWLEDGE AND CHECK VOLUME VALID

030772  013760  001416  000014          MOV     RMER10,RMER1(R0)        ;LOAD RMER1

031000  012760  000023  000000          MOV     #PACACK!GO,RMCS1(R0)    ;LOAD RMCS1
031006  012703  000003                  MOV     #3,R3
031012                          10$:

031012  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1

031020  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
031026  005303                          DEC     R3
031030  001370                          BNE     10$             ;ISSUE NEXT CLOCK IF COUNT NOT 0

031032  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
031040  042737  177677  001142          BIC     #↑CVV,$BDDAT
031046  001414                          BEQ     20$             ;BRANCH IF VV IS ZERO
031050  005737  001416                  TST     RMER10
031054  001527                          BEQ     70$             ;BRANCH IF VV SHOULD BE SET
031056  005037  001140                  CLR     $GDDAT          ;SETUP ERROR MESSAGE
031062  010037  001136                  MOV     R0,$BDADR
031066  062737  000012  001136          ADD     #RMDS,$BDADR
031074  104163                          ERROR   163             ;DECODE SET WITH COMP ERROR ACTIVE
031076  000522                          BR      80$             ;SKIP
031100                          20$:
031100  004737  031346                  JSR     PC,100$         ;INITIALIZE AND SET DIAGNOSTIC MODE

                                ;EXECUTE A READ IN PRESET AND CHECK VOLUME VALID

031104  013760  001416  000014          MOV     RMER10,RMER1(R0)        ;LOAD RMER1

031112  012760  000021  000000          MOV     #RIP!GO,RMCS1(R0)       ;LOAD RMCS1
031120  012703  000003                  MOV     #3,R3           ;R3=CLOCK COUNT
031124                          30$:

031124  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

031132  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
031140  005303                          DEC     R3
031142  001370                          BNE     30$             ;ISSUE NEXT CLOCK IF COUNT NOT ZERO

031144  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
031152  042737  177677  001142          BIC     #↑CVV,$BDDAT
031160  001414                          BEQ     40$             ;BRANCH IF VOLUME VALID NOT SET
031162  005737  001416                  TST     RMER10
031166  001462                          BEQ     70$             ;BRANCH IF VOLUME VALID SHOULD BE SET
031170  005037  001140                  CLR     $GDDAT          ;SETUP ERROR MESSAGE
```

```
031174  010037  001136                          MOV     R0,$BDADR
031200  062737  000012  001136                  ADD     #RMDS,$BDADR
031206  104163                                  ERROR   163                     ;DECODE SET WITH COMP ERROR ACTIVE
031210  000455                                  BR      80$                     ;SKIP
031212                                  40$:
031212  004737  031346                          JSR     PC,100$                 ;INITIALIZE AND SET DIAGNOSTIC MODE

                                        ;EXECUTE A WRITE CHECK DATA AND CHECK OCCUPIED

031216  013760  001416  000014                  MOV     RMER10,RMER1(R0)                ;LOAD RMER1

031224  012760  000051  000000                  MOV     #WCD!GO,RMCS1(R0)               ;LOAD RMCS1
031232  012703  000002                          MOV     #2,R3                   ;R3=CLOCK COUNT
031236                                  50$:

031236  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

031244  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
031252  005303                                  DEC     R3
031254  001370                                  BNE     50$                     ;ISSUE NEXT CLOCK IF COUNT NOT ZERO

031256  016037  000024  001142                  MOV     RMMR1(R0),$BDDAT                ;STORE RMMR1 AT $BDDAT
031264  042737  077777  001142                  BIC     #↑COCC,$BDDAT
031272  001414                                  BEQ     60$                     ;BRANCH IF OCCUPIED IS RESET
031274  005737  001416                          TST     RMER10
031300  001415                                  BEQ     70$                     ;BRANCH IF OCCUPIED SHOULD BE SET
031302  005037  001140                          CLR     $GDDAT                  ;SETUP ERROR MESSAGE
031306  010037  001136                          MOV     R0,$BDADR
031312  062737  000024  001136                  ADD     #RMMR1,$BDADR
031320  104164                                  ERROR   164                     ;DECODE SET WITH COMP ERROR ACTIVE
031322  000410                                  BR      80$
031324                                  60$:
                                        ;VOLUME VALID AND OCCUPIED DID NOT SET-SEE IF COMP ERROR WAS ACTIVE
031324  005737  001416                          TST     RMER10
031330  001005                                  BNE     80$                     ;BRANCH IF COMP ERROR WAS SET
                                        ;COULD NOT SET VV OR OCCUPIED-SUSPECT DECODE FLOP NOT SETTING
031332  104165                                  ERROR   165                     ;DECODE DOES NOT SET

031334                                  70$:

                                        ;REPEAT TEST WITH COMPOSITE ERROR ACTIVE-VERIFY THAT DECODE FLOP
                                        ;DOES NOT SET, AS INDICATED BY VOLUME VALID AND OCCUPIED.
031334  012737  040000  001416                  MOV     #UNS,RMER10             ;USE UNSAFE TO SET COMP ERROR
031342  000611                                  BR      5$

031344  000513                          80$:    BR      200$                    ;END OF TEST

031346                                  100$:

                                        ;SUBROUTINE USED DURING TEST

                                        ;USING DIAGNOSTIC MODE, RESET VOLUME VALID AND COMPOSITE ERROR.
                                        ;VERIFY THAT VV, ERR, AND OCC ARE ZERO.
031346  012760  000040  000010                  MOV     #CLR,RMCS2(R0)          ;CLEAR THE MASSBUS
031354  111160  000010                          MOVB    (R1),RMCS2(R0)          ;SELECT UNIT
```

```
031360  012760  000001  000024           MOV     #DMD,RMMR1(R0)   ;LOAD RMMR1

031366  012760  041001  000024           MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

031374  012760  000000  000014           MOV     #0,RMER1(R0)     ;LOAD RMER1

031402  012760  000000  000042           MOV     #0,RMER2(R0)     ;LOAD RMER2
031410  005037  001140                   CLR     $GDDAT           ;SETUP FOR ERROR MSG
031414  010037  001136                   MOV     R0,$BDADR
031420  062737  000012  001136           ADD     #RMDS,$BDADR

031426  016037  000012  001142           MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
031434  042737  137777  001142           BIC     #↑CERR,$BDDAT
031442  001402                           BEQ     110$             ;BRANCH IF COMP ERROR ZERO
031444  104143                           ERROR   143              ;CANT CLEAR COMP ERROR
031446  000447                           BR      140$             ;SKIP TEST
031450                           110$:

031450  016037  000012  001142           MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
031456  042737  177677  001142           BIC     #↑CVV,$BDDAT
031464  001402                           BEQ     120$             ;BRANCH IF VOLUME VALID ZERO
031466  104135                           ERROR   135              ;CANT RESET VOLUME VALID
031470  000436                           BR      140$             ;SKIP TEST
031472                           120$:

031472  016037  000024  001142           MOV     RMMR1(R0),$BDDAT         ;STORE RMMR1 AT $BDDAT
031500  042737  077777  001142           BIC     #↑COCC,$BDDAT
031506  001407                           BEQ     130$             ;BRANCH IF OCCUPIED ZERO
031510  010037  001136                   MOV     R0,$BDADR        ;SETUP ERROR MESSAGE
031514  062737  000024  001136           ADD     #RMMR1,$BDADR
031522  104166                           ERROR   166              ;CANT CLEAR OCCUPIED
031524  000420                           BR      140$             ;SKIP TEST
031526                           130$:
                                 ;TO VERIFY THAT THE DECODE FLOP IS RESET, LOAD AN ILLEGAL FUNCTION
                                 ;IN RMCS1 AND VERIFY THAT ILF DOES NOT SET.

031526  012760  000024  000000           MOV     #ILF24,RMCS1(R0)         ;LOAD RMCS1

031534  016037  000014  001142           MOV     RMER1(R0),$BDDAT         ;STORE RMER1 AT $BDDAT
031542  042737  177776  001142           BIC     #↑CILF,$BDDAT
031550  001410                           BEQ     150$             ;BRANCH IF ILF IS ZERO
031552  010037  001136                   MOV     R0,$BDADR        ;SETUP ERROR MESSAGE
031556  062737  000014  001136           ADD     #RMER1,$BDADR
031564  104167                           ERROR   167              ;DECODE FLOP APPEARS ON
031566  012716  031574           140$:   MOV     #200$,(SP)       ;DONT GO BACK TO TEST

031572  000207                   150$:   RTS     PC               ;RETURN TO TEST OR EXIT TEST

031574                           200$:

                                 ;;*****************************************************
                                 ;*TEST 53        SET/RESET VOLUME VALID TEST

                                 ;;*****************************************************
031574  000004                   TST53:  SCOPE
031576  012737  000053  001226           MOV     #53,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
```

```

031604  000240                                  NOP
031606  012737  000024  001120                  MOV     #20.,$ICNT              ;20 ITERATIONS
031614  112737  000001  001131                  MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
031622  012737  031636  001122                  MOV     #T53,$LPADR     ;LOAD LOOP ON TEST ADDRESS
031630  012737  031636  001124                  MOV     #T53,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
031636                                  T53:
031636  012706  001100                          MOV     #STACK,SP               ;LOAD THE STACK POINTER
031642  013700  001276                          MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
031646  013701  001456                          MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
031652  010037  001136                          MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS
031656  062737  000012  001136                  ADD     #RMDS,$BDADR
031664  012702  032076                          MOV     #100$,R2                ;R2=TABLE POINTER
031670                                  10$:
                                        ;INITIALIZE AND USE DIAGNOSTIC MODE TO RESET VOLUME VALID
031670  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
031676  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

031702  012760  000001  000024                  MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

031710  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

031716  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1

031724  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

031732  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
031740  042737  177677  001142                  BIC     #^CVV,$BDDAT
031746  001403                                  BEQ     20$                     ;BRANCH IF VOLUME VALID ZERO
031750  005037  001140                          CLR     $GDDAT
031754  104135                                  ERROR   135                     ;CANT RESET VV
031756                                  20$:
                                        ;EXECUTE THE FUNCTION CODE IN THE TABLE
031756  111203                                  MOVB    (R2),R3                 ;GET FUNCTION CODE
031760  042703  177701                          BIC     #^CILF76,R3
031764  052703  000001                          BIS     #GO,R3

031770  010360  000000                          MOV     R3,RMCS1(R0)            ;LOAD RMCS1
031774  116204  000001                          MOVB    1(R2),R4                ;GET CLOCK COUNT
032000  042704  177400                          BIC     #^C377,R4
032004                                  30$:

032004  012760  141001  000024                  MOV     #DMD!DBEN!MUR!DBCK,RMMR1(R0)    ;LOAD RMMR1

032012  012760  041001  000024                  MOV     #DMD!DBEN!MUR,RMMR1(R0) ;LOAD RMMR1
032020  005304                                  DEC     R4
032022  001370                                  BNE     30$                     ;ISSUE COCKS TIL R4 ZERO

032024  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
032032  042737  177677  001142                  BIC     #^CVV,$BDDAT
032040  001007                                  BNE     40$                     ;BRANCH IF VOLUME VALID SET
032042  010337  001174                          MOV     R3,$TMP0                ;SAVE FUNCTION CODE FOR MSG
032046  012737  000100  001140                  MOV     #VV,$GDDAT
032054  104170                                  ERROR   170                     ;CANT SET VOLUME VALID
032056  000406                                  BR      50$
032060                                  40$:
```

```
                                        ;ADVANCE THE TABLE POINTER, EXIT IF DONE
032060  062702  000002                          ADD     #2,R2
032064  105762  000001                          TSTB    1(R2)
032070  100401                                  BMI     50$                     ;EXIT IF COUNT IS NEGATIVE
032072  000676                                  BR      10$
032074  000403                          50$:    BR      200$                    ;JUMP OVER TABLE

032076                                  100$:
                                        ;TABLE OF FUNCTION CODES AND CLOCK COUNTS
032076     020                                  .BYTE   RIP                     ;READ IN PRESET COMMAND
032077     003                                  .BYTE   3

032100     022                                  .BYTE   PAKACK                  ;PACK ACKNOWLEDGE COMMAND
032101     003                                  .BYTE   3

032102     000                                  .BYTE                           ;END OF TABLE
032103     377                                  .BYTE   -1

032104                                  200$:                                   ;END OF TEST


                                        ;;*****************************************************************
                                        ;*TEST 54               ILLEGAL FUNCTION TEST

                                        ;;*****************************************************************
032104  000004                          TST54:  SCOPE
032106  012737  000054  001226                  MOV     #54,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

032114  000240                                  NOP
032116  012737  000024  001120                  MOV     #20.,$ICNT              ;20 ITERATIONS
032124  112737  000001  001131                  MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
032132  012737  032146  001122                  MOV     #T54,$LPADR     ;LOAD LOOP ON TEST ADDRESS
032140  012737  032146  001124                  MOV     #T54,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
032146                                  T54:
032146  012706  001100                          MOV     #STACK,SP               ;LOAD THE STACK POINTER
032152  013700  001276                          MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
032156  013701  001456                          MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
032162  005002                                  CLR     R2                      ;INITIALIZE FUNCTION CODE VALUE
032164                                  10$:
                                        ;SET VOLUME VALID USING SUBROUTINE
032164  004737  060526                          JSR     PC,SETVV                ;GO SET VOLUME VALID
032170  000402                                  BR      20$                     ;BRANCH TO 20$ IF NO ERROR
032172  104000                                  ERROR                           ;RETURN HERE IF ERROR
032174  000460                                  BR      50$                     ;SKIP TEST IF ERROR
032176  012704  000002                  20$:    MOV     #2,R4                   ;R4=CLOCK COUNT
                                        ;EXECUTE THE TEST FUNCTION CODE AND VERIFY ILF

032202  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
032210  010203                                  MOV     R2,R3                   ;SETUP FUNCTION CODE IN R3
032212  052703  000001                          BIS     #GO,R3

032216  010360  000000                          MOV     R3,RMCS1(R0)            ;LOAD RMCS1
032222                                  30$:

032222  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

032230  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
```

```
032236  005304                          DEC     R4
032240  001370                          BNE     30$

032242  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
032250  042737  177776  001142          BIC     #↑CILF,$BDDAT   ;SETUP ACTUAL ILF STATUS
032256  016237  065764  001140          MOV     FNCDTB(R2),$GDDAT  ;GET EXPECTED ILF STATUS
032264  042737  177776  001140          BIC     #↑CILF,$GDDAT
032272  023737  001140  001142          CMP     $GDDAT,$BDDAT
032300  001410                          BEQ     40$                     ;BRANCH IF ILF IS OK
032302  010037  001136                  MOV     R0,$BDADR               ;SETUP FOR ERROR MSG
032306  062737  000014  001136          ADD     #RMER1,$BDADR
032314  010237  001174                  MOV     R2,$TMP0
032320  104171                          ERROR   171                     ;ILF IS NOT CORRECT
032322                          40$:
                                ;ADVANCE TO THE NEXT FUNCTION CODE AND REPEAT TEST
032322  062702  000002                  ADD     #2,R2
032326  022702  000076                  CMP     #ILF76,R2
032332  103401                          BLO     50$
032334  000713                          BR      10$
032336                          50$:                                    ;END OF TEST


                                ;;*****************************************************************
                                ;*TEST 55               OCCUPIED TEST

                                ;;*****************************************************************
032336  000004                  TST55:  SCOPE
032340  012737  000055  001226          MOV     #55,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

032346  000240                          NOP
032350  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
032356  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
032364  012737  032400  001122          MOV     #T55,$LPADR     ;LOAD LOOP ON TEST ADDRESS
032372  012737  032400  001124          MOV     #T55,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
032400                          T55:
032400  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
032404  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
032410  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
032414  005002                          CLR     R2                              ;INITIALIZE FUNCTION CODE

032416                          10$:
                                ;GET THE DEVICE READY
                                ;SET VOLUME VALID USING SUBROUTINE
032416  004737  060526                  JSR     PC,SETVV                ;GO SET VOLUME VALID
032422  000402                          BR      20$                     ;BRANCH TO 20$ IF NO ERROR
032424  104000                          ERROR                           ;RETURN HERE IF ERROR
032426  000464                          BR      50$
032430                          20$:
                                ;ENABLE DEBUG CLOCK AND LOAD THE FUNCTION CODE

032430  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
032436  010203                          MOV     R2,R3                           ;ASSEMBLE FUNCTION CODE AND
032440  052703  000001                  BIS     #GO,R3                          ;GO BIT IN R3

032444  010360  000000                  MOV     R3,RMCS1(R0)            ;LOAD RMCS1
032450  012704  000002                  MOV     #2,R4                           ;R4=CLOCK COUNT
                                ;STEP THE DEBUG CLOCK UNTIL SET PULSE IS ACTIVE
```

```
032454                              30$:
032454  012760  141001  000024              MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
032462  012760  041001  000024              MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
032470  005304                              DEC     R4
032472  001370                              BNE     30$                     ;ISSUE NEXT CLOCK TIL R4 ZERO
                                    ;VERIFY OCCUPIED STATUS
032474  016037  000024  001142              MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
032502  042737  077777  001142              BIC     #↑COCC,$BDDAT
032510  005037  001140                      CLR     $GDDAT          ;GENERATE OCC FROM AOE
032514  032762  001000  065764              BIT     #AOE,FNCDTB(R2)
032522  001403                              BEQ     35$
```

```
032524  012737  100000  001140          MOV     #OCC,$GDDAT
032532  023737  001140  001142  35$:    CMP     $GDDAT,$BDDAT
032540  001411                          BEQ     40$                     ;BRANCH IF OCC IS OK
032542  010237  001174                  MOV     R2,$TMPO                ;SAVE FUNCTION CODE
032546  010037  001136                  MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS
032552  062737  000024  001136          ADD     #RMMR1,$BDADR
032560  104173                          ERROR   173                     ;OCCUPIED IS INCORRECT
032562  000406                          BR      50$
032564                          40$:
                                ;ADVANCE TO NEXT FUNCTIONCODE, EXIT IF DONE
032564  062702  000002                  ADD     #2,R2
032570  022702  000076                  CMP     #ILF76,R2
032574  103401                          BLO     50$                     ;EXIT IF DONE
032576  000707                          BR      10$

032600                          50$:                                    ;END OF TEST

                                ;;*********************************************************
                                ;*TEST 56        READ IN PRESET TEST

                                ;;*********************************************************
032600  000004                  TST56:  SCOPE
032602  012737  000056  001226          MOV     #56,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

032610  000240                          NOP
032612  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
032620  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
032626  012737  032642  001122          MOV     #T56,$LPADR     ;LOAD LOOP ON TEST ADDRESS
032634  012737  032642  001124          MOV     #T56,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
032642                          T56:
032642  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
032646  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
032652  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                ;CLEAR AND ENABLE DEBUG CLOCK - LEAVE VOLUME VALID RESET
032656  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
032664  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

032670  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

032676  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

032704  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

032712  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
                                ;LOAD ALL ONES IN RMDA, RMDC AND RMOF

032720  012760  177777  000006          MOV     #-1,RMDA(R0)    ;LOAD RMDA

032726  012760  177777  000034          MOV     #-1,RMDC(R0)    ;LOAD RMDC

032734  012760  177777  000032          MOV     #-1,RMOF(R0)    ;LOAD RMOF
                                ;LOAD READ IN PRESET COMMAND AND STEP THE CLOCK TILL SET PULSE

032742  012760  000021  000000          MOV     #RIP!GO,RMCS1(R0)       ;LOAD RMCS1
032750  012702  000003                  MOV     #3,R2                   ;R2=CLOCK COUNT
032754                          10$:
```

```
032754 012760 141001 000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
032762 012760 041001 000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
032770 005302                        DEC     R2
032772 001370                        BNE     10$                     ;ISSUE 3 CLOCKS
                             ;SEE IF RMDA OR RMDC OR RMOF IS ZERO
032774 016002 000006                 MOV     RMDA(R0),R2     ;STORE RMDA AT R2
033000 005702                        TST     R2
033002 001413                        BEQ     20$                     ;BRANCH IF RMDA IS ZERO
033004 016002 000034                 MOV     RMDC(R0),R2     ;STORE RMDC AT R2
033010 042702 176000                 BIC     #XNUDC,R2               ;CLEAR UNUSED BITS
033014 001406                        BEQ     20$                     ;BRANCH IF RMDC IS ZERO
033016 016002 000032                 MOV     RMOF(R0),R2     ;STORE RMOF AT R2
033022 042702 161577                 BIC     #XNUOF,R2               ;CLEAR UNUSED BITS
033026 001401                        BEQ     20$                     ;BRANCH IF RMOF IS ZERO
                             ;READ IN PRESET COMMAND DIDNT CLEAR ANY OF THE 3 REGISTERS
033030 104174                        ERROR   174                     ;READ IN PRESET FAILED
033032                       20$:                                    ;END OF TEST
                             ;;*************************************************************
                             ;*TEST 57        RIP/RMOF TEST
                             ;;*************************************************************
033032 000004                TST57:  SCOPE
033034 012737 000057 001226         MOV     #57,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
033042 000240                        NOP
033044 012737 000024 001120          MOV     #20.,$ICNT      ;20 ITERATIONS
033052 112737 000001 001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
033060 012737 033074 001122          MOV     #T57,$LPADR ;LOAD LOOP ON TEST ADDRESS
033066 012737 033074 001124          MOV     #T57,$LPERR ;LOAD LOOP ON ERROR ADDRESS
033074                       T57:
033074 012706 001100                 MOV     #STACK,SP       ;LOAD THE STACK POINTER
033100 013700 001276                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
033104 013701 001456                 MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
033110 010037 001136                 MOV     R0,$BDADR               ;SETUP REGISTER ADDRESS AND
033114 062737 000032 001136          ADD     #RMOF,$BDADR
033122 005037 001140                 CLR     $GDDAT                  ;EXPECTED RMOF
                             ;INITIALIZE AND SET BITS IN RMOF
033126 012760 000040 000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
033134 111160 000010                 MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
033140 012760 000001 000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
033146 012760 041001 000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
033154 012760 000000 000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
033162 012760 000000 000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
```

```
033170  012760  177777  000032          MOV     #-1,RMOF(R0)     ;LOAD RMOF
                                ;EXECUTE A READ IN PRESET IN DIAGNOSTIC MODE TILL SET PULSE

033176  012760  000021  000000          MOV     #RIP!GO,RMCS1(R0)               ;LOAD RMCS1
033204  012702  000003                  MOV     #3,R2                           ;R2=CLOCK COUNT
033210                          10$:

033210  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1

033216  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
033224  005302                          DEC     R2
033226  001370                          BNE     10$                             ;ISSUE 3 CLOCKS
                                ;VERIFY THAT RMOF IS ZERO

033230  016037  000032  001142          MOV     RMOF(R0),$BDDAT ;STORE RMOF AT $BDDAT
033236  042737  161577  001142          BIC     #XNUOF,$BDDAT
033244  001401                          BEQ     20$                             ;BRANCH IF RMOF IS ZERO
033246  104175                          ERROR   175                             ;CANT CLEAR RMOF WITH RIP

033250                          20$:                                            ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 60        RMDA/RMDC/RIP TEST

                                ;;*****************************************************************
033250  000004                  TST60:  SCOPE
033252  012737  000060  001226          MOV     #60,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

033260  000240                          NOP
033262  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
033270  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
033276  012737  033312  001122          MOV     #T60,$LPADR     ;LOAD LOOP ON TEST ADDRESS
033304  012737  033312  001124          MOV     #T60,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
033312                          T60:
033312  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
033316  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
033322  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
033326  005037  001140                  CLR     $GDDAT

                                ;CLEAR, ENABLE DEBUG CLOCK, THEN PRESET RMDA AND RMDC
033332  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
033340  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

033344  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

033352  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

033360  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

033366  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

033374  012760  177777  000006          MOV     #-1,RMDA(R0)    ;LOAD RMDA

033402  012760  177777  000034          MOV     #-1,RMDC(R0)    ;LOAD RMDC
                                ;EXECUTE READ IN PRESET TILL SET PULSE
```

```
033410  012760  000021  000000          MOV     #RIP!GO,RMCS1(R0)       ;LOAD RMCS1
033416  012702  000003                  MOV     #3,R2
033422                          10$:

033422  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

033430  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
033436  005302                          DEC     R2
033440  001370                          BNE     10$                     ;ISSUE 3 CLOCKS
                                ;VERIFY RMDA IS ZERO

033442  016037  000006  001142          MOV     RMDA(R0),$BDDAT ;STORE RMDA AT $BDDAT
033450  005737  001142                  TST     $BDDAT
033454  001406                          BEQ     20$                     ;BRANCH IF RMDA RESET
033456  010037  001136                  MOV     R0,$BDADR
033462  062737  000006  001136          ADD     #RMDA,$BDADR
033470  104176                          ERROR   176                     ;RMDA NOT RESET BY RIP
033472                          20$:
                                ;VERIFY RMDC IS ZERO

033472  016037  000034  001142          MOV     RMDC(R0),$BDDAT ;STORE RMDC AT $BDDAT
033500  042737  176000  001142          BIC     #XNUDC,$BDDAT
033506  001406                          BEQ     30$                     ;BRANCH IF RMDC RESET
033510  010037  001136                  MOV     R0,$BDADR
033514  062737  000034  001136          ADD     #RMDC,$BDADR
033522  104177                          ERROR   177                     ;RMDC NOT RESET BY RIP

033524                          30$:                                    ;END OF TEST

                                ;;*************************************************************
                                ;*TEST 61       OFFSET COMMAND TEST

                                ;;*************************************************************
033524  000004                  TST61:  SCOPE
033526  012737  000061  001226          MOV     #61,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

033534  000240                          NOP
033536  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
033544  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
033552  012737  033566  001122          MOV     #T61,$LPADR     ;LOAD LOOP ON TEST ADDRESS
033560  012737  033566  001124          MOV     #T61,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
033566                          T61:
033566  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
033572  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
033576  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
033602  010037  001136                  MOV     R0,$BDADR
033606  062737  000012  001136          ADD     #RMDS,$BDADR
033614  012737  000001  001140          MOV     #0M,$GDDAT


                                ;SET VOLUME VALID USING SUBROUTINE
033622  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
033626  000402                          BR      10$             ;BRANCH TO 10$ IF NO ERROR
033630  104000                          ERROR                   ;RETURN HERE IF ERROR
033632  000433                          BR      40$
033634                          10$:
```

```
033634  012760  000000  000034          MOV     #0,RMDC(R0)             ;LOAD RMDC
                                ;ENABLE DEBUG CLOCK AND EXECUTE OFFSET COMMAND

033642  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

033650  012760  000015  000000          MOV     #OFFSET!GO,RMCS1(R0)    ;LOAD RMCS1
033656  012702  000002                  MOV     #2,R2                   ;R2=CLOCK COUNT
033662                          20$:

033662  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

033670  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
033676  005302                          DEC     R2
033700  001370                          BNE     20$                     ;ISSUE 2 CLOCKS
                                ;VERIFY THAT OFFSET MODE IS SET

033702  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
033710  042737  177776  001142          BIC     #↑COM,$BDDAT
033716  001001                          BNE     40$                     ;BRANCH IF OM IS SET
033720  104200                          ERROR   200                     ;CANT SET ON BY COMMAND
033722                          40$:                                    ;END OF TEST


                                ;;*****************************************************************
                                ;*TEST 62        RETURN TO CENTER TEST

                                ;;*****************************************************************
033722  000004                  TST62:  SCOPE
033724  012737  000062  001226          MOV     #62,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

033732  000240                          NOP
033734  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
033742  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
033750  012737  033764  001122          MOV     #T62,$LPADR     ;LOAD LOOP ON TEST ADDRESS
033756  012737  033764  001124          MOV     #T62,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
033764                          T62:
033764  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
033770  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
033774  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
034000  010037  001136                  MOV     R0,$BDADR
034004  062737  000012  001136          ADD     #RMDS,$BDADR
                                ;SET OFFSET DIRECTION AND OFFSET MODE
                                ;SET VOLUME VALID USING SUBROUTINE
034012  004737  060526                  JSR     PC,SETVV                ;GO SET VOLUME VALID
034016  000402                          BR      10$                     ;BRANCH TO 10$ IF NO ERROR
034020  104000                          ERROR                           ;RETURN HERE IF ERROR
034022  000465                          BR      60$
034024                          10$:

034024  012760  000200  000032          MOV     #OFD,RMOF(R0)           ;LOAD RMOF
                                ;SET OFFSET MODE USING SUBROUTINE
034032  004737  060656                  JSR     PC,SETOM                ;GO SET OFFSET MODE
034036  000401                          BR      20$                     ;BRANCH TO 20$ IF NO ERROR
034040  104000                          ERROR                           ;RETURN HERE IF ERROR
034042                          20$:
                                ;ENABLE DEBUG CLOCK AND EXECUTE RETURN TOCENTER COMMAND
                                ;SET VOLUME VALID USING SUBROUTINE
```

```
034042  004737  060526                  JSR     PC,SETVV                ;GO SET VOLUME VALID
034046  000402                          BR      30$                     ;BRANCH TO 30$ IF NO ERROR
034050  104000                          ERROR                           ;RETURN HERE IF ERROR
034052  000451                          BR      60$
034054                          30$:

034054  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

034062  012760  000017  000000          MOV     #RTC!GO,RMCS1(R0)       ;LOAD RMCS1
034070  012702  000002                  MOV     #2,R2
034074                          40$:

034074  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

034102  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
034110  005302                          DEC     R2
034112  001370                          BNE     40$                     ;ISSUE 2 CLOCKS
                                ;VERIFY THAT OFFSET MODE IS RESET

034114  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
034122  042737  177776  001142          BIC     #↑COM,$BDDAT
034130  001403                          BEQ     50$                     ;BRANCH IF OFFSET MODE RESET
034132  005037  001140                  CLR     $GDDAT
034136  104201                          ERROR   201                     ;CANT RESET OFFSETMODE BY RTC
034140                          50$:
                                ;VERIFY THAT OFFSET DIRECTION IS RESET

034140  016037  000032  001142          MOV     RMOF(R0),$BDDAT ;STORE RMOF AT $BDDAT
034146  042737  177577  001142          BIC     #↑COFD,$BDDAT
034154  001410                          BEQ     60$                     ;BRANCH IF OFD IS RESET
034156  005037  001140                  CLR     $GDDAT
034162  010037  001136                  MOV     R0,$BDADR
034166  062737  000032  001136          ADD     #RMOF,$BDADR
034174  104202                          ERROR   202                     ;CANT RESET OFD BY RTC
034176                          60$:                                    ;END OF TEST

                                ;;*****************************************************
                                ;*TEST 63        RMDC CLEAR OFFSET TEST

                                ;;*****************************************************
034176  000004                  TST63:  SCOPE
034200  012737  000063  001226          MOV     #63,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

034206  000240                          NOP
034210  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
034216  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
034224  012737  034240  001122          MOV     #T63,$LPADR     ;LOAD LOOP ON TEST ADDRESS
034232  012737  034240  001124          MOV     #T63,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
034240                          T63:
034240  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
034244  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
034250  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
034254  010037  001136                  MOV     R0,$BDADR
034260  062737  000012  001136          ADD     #RMDS,$BDADR
                                ;SET VOLUME VALID USING SUBROUTINE
034266  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
```

```
034272  000402                                  BR      10$                     ;BRANCH TO 10$ IF NO ERROR
034274  104000                                  ERROR                           ;RETURN HERE IF ERROR
034276  000421                                  BR      40$                             ;SKIP REST OF TEST
034300                                  10$:
                                        ;SET OFFSET MODE USING SUBROUTINE
034300  004737  060656                          JSR     PC,SETOM                ;GO SET OFFSET MODE
034304  000401                                  BR      20$                     ;BRANCH TO 20$ IF NO ERROR
034306  104000                                  ERROR                           ;RETURN HERE IF ERROR
034310                                  20$:
                                        ;WRITE THE DESIRED CYLINDER REGISTER AND VERIFY THAT OFFSET IS ZERO

034310  012760  000000  000034                  MOV     #0,RMDC(R0)             ;LOAD RMDC

034316  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
034324  042737  177776  001142                  BIC     #↑COM,$BDDAT
034332  001403                                  BEQ     40$
034334  005037  001140                          CLR     $GDDAT
034340  104203                                  ERROR   203                             ;CANT RESET OFFSET BY RMDC

034342                                  40$:                                            ;END OF TEST

                                        ;;*************************************************************
                                        ;*TEST 64               EBL CLEAR OFFSET TEST

                                        ;;*************************************************************
034342  000004                          TST64:  SCOPE
034344  012737  000064  001226                  MOV     #64,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX

034352  000240                                  NOP
034354  012737  000024  001120                  MOV     #20.,$ICNT              ;20 ITERATIONS
034362  112737  000001  001131                  MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
034370  012737  034404  001122                  MOV     #T64,$LPADR     ;LOAD LOOP ON TEST ADDRESS
034376  012737  034404  001124                  MOV     #T64,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
034404                                  T64:
034404  012706  001100                          MOV     #STACK,SP               ;LOAD THE STACK POINTER
034410  013700  001276                          MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
034414  013701  001456                          MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
034420  010037  001136                          MOV     R0,$BDADR                       ;SETUP REGISTER FOR ERROR MSG
034424  062737  000012  001136                  ADD     #RMDS,$BDADR
                                        ;SET VOLUME VALID USING SUBROUTINE
034432  004737  060526                          JSR     PC,SETVV                ;GO SET VOLUME VALID
034436  000402                                  BR      10$                     ;BRANCH TO 10$ IF NO ERROR
034440  104000                                  ERROR                           ;RETURN HERE IF ERROR
034442  000440                                  BR      30$                             ;SKIP REST OF TEST IF ERROR
034444                                  10$:
                                        ;SET OFFSET MODE USING SUBROUTINE
034444  004737  060656                          JSR     PC,SETOM                ;GO SET OFFSET MODE
034450  000401                                  BR      20$                     ;BRANCH TO 20$ IF NO ERROR
034452  104000                                  ERROR                           ;RETURN HERE IF ERROR
034454                                  20$:
                                        ;SET 16 BIT FORMAT AND LOAD LAST SECTOR/TRACK ADDRESS

034454  012760  010000  000032                  MOV     #FMT16,RMOF(R0) ;LOAD RMOF

034462  012760  002037  000006                  MOV     #002037,RMDA(R0)                ;LOAD RMDA
                                        ;FORCE END OF BLOCK AND VERIFY THAT OFFSET MODE IS CLEARED
```

```
034470 012760 041001 000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

034476 012760 000001 000000          MOV     #GO,RMCS1(R0)     ;LOAD RMCS1

034504 012760 061001 000024          MOV     #DMD!MUR!DBEN!DEBL,RMMR1(R0)     ;LOAD RMMR1

034512 012760 041001 000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

034520 016037 000012 001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
034526 042737 177776 001142          BIC     #↑COM,$BDDAT
034534 001403                        BEQ     30$                     ;BRANCH IF OFFSET IS ZERO
034536 005037 001140                 CLR     $GDDAT
034542 104204                        ERROR   204                     ;OFFSET NO CLEARED BY EBL
034544                       30$:                                    ;END OF TEST

                             ;;*****************************************************************
                             ;*TEST 65        RUN AND GO TEST

                             ;;*****************************************************************
034544 000004                TST65:  SCOPE
034546 012737 000065 001226          MOV     #65,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
034554 000240                        NOP
034556 012737 000024 001120          MOV     #20.,$ICNT        ;20 ITERATIONS
034564 112737 000001 001131          MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
034572 012737 034606 001122          MOV     #T65,$LPADR  ;LOAD LOOP ON TEST ADDRESS
034600 012737 034606 001124          MOV     #T65,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
034606                       T65:
034606 012706 001100                 MOV     #STACK,SP         ;LOAD THE STACK POINTER
034612 013700 001276                 MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
034616 013701 001456                 MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
034622 005037 001140                 CLR     $GDDAT                  ;INITIALIZE EXPECTED RESULT
034626 005002                        CLR     R2                      ;INTIALIZE FUNCTION CODE
034630 010037 001136                 MOV     R0,$BDADR
034634 062737 000024 001136          ADD     #RMMR1,$BDADR
034642                       10$:
                             ;CLEAR THE MASSBUS AND ENABLE DEBUG CLOCK
034642 012760 000040 000010          MOV     #CLR,RMCS2(R0)    ;CLEAR THE MASSBUS
034650 111160 000010                 MOVB    (R1),RMCS2(R0)    ;SELECT UNIT

034654 012760 000001 000024          MOV     #DMD,RMMR1(R0)    ;LOAD RMMR1

034662 012760 040001 000024          MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1

034670 012760 000000 000014          MOV     #0,RMER1(R0)      ;LOAD RMER1

034676 012760 000000 000042          MOV     #0,RMER2(R0)      ;LOAD RMER2
                             ;LOAD THE FUNCTION CODE AND VERIFY RUN AND GO FLOP

034704 010203                        MOV     R2,R3                   ;ASSEMBLE FUNCTION CODE AND GO
034706 052703 000001                 BIS     #GO,R3
034712 012737 000200 001524          MOV     #200,WATCH        ;SET WATCHDOG TIMER VALUE
034720 004777 144602                 JSR     PC,@CLOCK         ;START THE CLOCK

034724 010360 000000                 MOV     R3,RMCS1(R0)      ;LOAD RMCS1
034730                       15$:
```

```

034730  016037  000024  001142          MOV     RMMR1(R0),$BDDAT             ;STORE RMMR1 AT $BDDAT
034736  042737  137777  001142          BIC     #↑CRG,$BDDAT
034744  023737  001140  001142          CMP     $GDDAT,$BDDAT
034752  001411                          BEQ     20$                          ;BRANCH IF RUN AND GO FLOP OK
034754  005737  001524                  TST     WATCH               ;TAKE ANOTHER SAMPLE IF CLOCK NOT ZERO
034760  001363                          BNE     15$
034762  004777  144542                  JSR     PC,@STOP            ;STOP THE CLOCK
034766  010237  001174                  MOV     R2,$TMP0                     ;SAVE FUNCTION CODE FOR MSG
034772  104205                          ERROR   205                          ;RUN AND GO INCORRECT
034774  000416                          BR      40$                          ;SKIP REST OF
034776                          20$:
034776  004777  144526                  JSR     PC,@STOP            ;STOP THE CLOCK
                                ;ADVNACE TO NEXT FUNCTION CODE - EXIT IF DONE
035002  062702  000002                  ADD     #2,R2
035006  022702  000076                  CMP     #ILF76,R2
035012  103407                          BLO     40$                          ;EXIT IF DONE
035014  020227  000050                  CMP     R2,#WCD                      ;CHANGE EXPECTED RESULTI IF
035020  103403                          BLO     30$                          ;DATA COMMAND
035022  012737  040000  001140          MOV     #RG,$GDDAT
035030  000704                  30$:    BR      10$                          ;REPEAT TEST

035032                          40$:                                         ;END OF TEST
                                ;;*****************************************************************
                                ;*TEST 66        SET IAE TEST

                                ;;*****************************************************************
035032  000004                  TST66:  SCOPE
035034  012737  000066  001226          MOV     #66,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX

035042  000240                          NOP
035044  012737  000024  001120          MOV     #20.,$ICNT          ;20 ITERATIONS
035052  112737  000001  001131          MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
035060  012737  035074  001122          MOV     #T66,$LPADR  ;LOAD LOOP ON TEST ADDRESS
035066  012737  035074  001124          MOV     #T66,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
035074                          T66:
035074  012706  001100                  MOV     #STACK,SP           ;LOAD THE STACK POINTER
035100  013700  001276                  MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
035104  013701  001456                  MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
035110  012702  035256                  MOV     #100$,R2                     ;R2=TABLE POINTER
035114                          10$:
                                ;SET VOLUME VALID USING SUBROUTINE
035114  004737  060526                  JSR     PC,SETVV            ;GO SET VOLUME VALID
035120  000402                          BR      20$                 ;BRANCH TO 20$ IF NO ERROR
035122  104000                          ERROR                       ;RETURN HERE IF ERROR
035124  000453                          BR      50$                          ;SKIP REST OF TEST
035126                          20$:
                                ;LOAD INVALID TRACK, SECTOR AND CYLINDER ADDRESS AND SET FORMAT TO 18

035126  012760  177777  000006          MOV     #-1,RMDA(R0)        ;LOAD RMDA

035134  012760  177777  000034          MOV     #-1,RMDC(R0)        ;LOAD RMDC

035142  012760  000000  000032          MOV     #0,RMOF(R0)         ;LOAD RMOF
                                ;ENABLE DEBUG CLOCK AND LOAD FUNCTION CODE IN RMCSI WITH GO ON

```

```
035150  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
035156  111203                          MOVB    (R2),R3
035160  042703  177701                  BIC     #↑CILF76,R3
035164  052703  000001                  BIS     #GO,R3

035170  010360  000000                  MOV     R3,RMCS1(R0)    ;LOAD RMCS1
035174  116204  000001                  MOVB    1(R2),R4                ;GET CLOCK COUNT
035200  042704  177400                  BIC     #↑C377,R4
035204                          30$:
                                ;CLOCK THE COMMAND SEQUENCER

035204  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

035212  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
035220  005304                          DEC     R4
035222  001370                          BNE     30$
                                ;SEE IF IAE HAS SET

035224  016004  000014                  MOV     RMER1(R0),R4    ;STORE RMER1 AT R4
035230  042704  175777                  BIC     #↑CIAE,R4
035234  001007                          BNE     50$                     ;BRANCH IF IAE SET
                                ;IAE DID NOT SET - TRY ANOTHER FUNCTION CODE
035236  062702  000002                  ADD     #2,R2
035242  105762  000001                  TSTB    1(R2)
035246  100401                          BMI     40$                     ;BRANCH IF ALL CODES TRIED
035250  000721                          BR      10$
035252                          40$:
                                ;CANNOT SET IAE WITH ANY COMBINATION OF ADDRESS AND FUNCTION CODE
035252  104206                          ERROR   206
035254                          50$:
035254  000411                          BR      200$                    ;JUMP OVER TABLE

035256                          100$:
                                ;TABLE OF FUNCTION CODES AND CLOCK COUNTS FOR TEST

035256     030                          .BYTE   SEARCH                  ;SEARCH COMMAND
035257     002                          .BYTE   2

035260     004                          .BYTE   SEEK                    ;SEEK COMMAND
035261     002                          .BYTE   2

035262     062                          .BYTE   WH                      ;WRITE HEADER COMMAND
035263     002                          .BYTE   2

035264     052                          .BYTE   WCH                     ;WRITE CHECK HEADER COMMAND
035265     002                          .BYTE   2

035266     072                          .BYTE   RH                      ;READ HEADER COMMAND
035267     002                          .BYTE   2

035270     060                          .BYTE   WD                      ;WRITE DATA COMMAND
035271     002                          .BYTE   2

035272     050                          .BYTE   WCD                     ;WRITE CHECK DATA COMMAND
035273     002                          .BYTE   2
```

```
035274	070				.BYTE	RD			;READ DATA COMMAND
035275	002				.BYTE	2

035276	000				.BYTE				;END OF TABLE
035277	377				.BYTE	-1

035300				200$:					;END OF TEST

				;;*************************************************************
				;*TEST 67		SEARCH, SEEK, READ WRITE TEST

				;;*************************************************************
035300	000004			TST67:	SCOPE
035302	012737	000067	001226		MOV	#67,$TESTN		;;SET TEST NUMBER IN APT MAIL BOX

035310	000240				NOP
035312	012737	000024	001120		MOV	#20.,$ICNT		;20 ITERATIONS
035320	112737	000001	001131		MOVB	#1,$ERMAX		;ONE ERROR ALLOWED
035326	012737	035342	001122		MOV	#T67,$LPADR	;LOAD LOOP ON TEST ADDRESS
035334	012737	035342	001124		MOV	#T67,$LPERR	;LOAD LOOP ON ERROR ADDRESS
035342				T67:
035342	012706	001100			MOV	#STACK,SP		;LOAD THE STACK POINTER
035346	013700	001276			MOV	$BASE,R0		;R0 = UNIBUS ADDRESS OF UUT
035352	013701	001456			MOV	TSTQUE,R1		;R1 = POINTER TO DEVICE
035356	005002				CLR	R2			;INITIALIZE FUNCTION CODE
035360				10$:
				;SET VOLUME VALID USING SUBROUTINE
035360	004737	060526			JSR	PC,SETVV		;GO SET VOLUME VALID
035364	000402				BR	20$			;BRANCH TO 20$ IF NO ERROR
035366	104000				ERROR				;RETURN HERE IF ERROR
035370	000472				BR	50$
035372				20$:
				;LOAD INVALID TRACK, SECTOR AND CYLINDER ADDRESS AND SET FORMAT
				;TO 18 BIT MODE

035372	012760	177777	000006		MOV	#-1,RMDA(R0)		;LOAD RMDA

035400	012760	177777	000034		MOV	#-1,RMDC(R0)		;LOAD RMDC

035406	012760	000000	000032		MOV	#0,RMOF(R0)		;LOAD RMOF
				;ENABLE DEBUG CLOCK AND LOAD FUNCTION CODE IN RMCS1 WITH GO ON

035414	012760	041001	000024		MOV	#DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
035422	010203				MOV	R2,R3			;ASSEMBLE CODE AND GO
035424	052703	000001			BIS	#GO,R3

035430	010360	000000			MOV	R3,RMCS1(R0)		;LOAD RMCS1
				;CLOCK THE COMMAND SEQUENCER TO SET PULSE
035434	012704	000002			MOV	#2,R4
035440				30$:

035440	012760	141001	000024		MOV	#DMD!MUR!DBEN!DBCK,RMMR1(R0)	;LOAD RMMR1

035446	012760	041001	000024		MOV	#DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
035454	005304				DEC	R4
035456	001370				BNE	30$
```

```
                                          ;VERIFY IAE ACCORDING TO FUNCTION CODE TABLE
035460  016037  000014  001142                    MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
035466  042737  175777  001142                    BIC     #↑CIAE,$BDDAT
035474  016237  065764  001140                    MOV     FNCDTB(R2),$GDDAT       ;ASSEMBLE EXPECTED IAE
035502  042737  175777  001140                    BIC     #↑CIAE,$GDDAT
035510  023737  001140  001142                    CMP     $GDDAT,$BDDAT
035516  001411                                    BEQ     40$                     ;BRANCH IF IAE OK
035520  010037  001136                            MOV     R0,$BDADR               ;SET UP ERROR MSG
035524  062737  000014  001136                    ADD     #RMER1,$BDADR
035532  010237  001174                            MOV     R2,$TMP0
035536  104207                                    ERROR   207                     ;IAE IS INCORRECT
035540  000406                                    BR      50$                     ;SKIP REST OF TEST
035542                                    40$:
                                          ;ADVANCE TO NEXT FUNCTIONCODE - EXIT IF DONE
035542  062702  000002                            ADD     #2,R2
035546  023702  000076                            CMP     ILF76,R2
035552  103401                                    BLO     50$
035554  000701                                    BR      10$
035556                                    50$:                                    ;END OF TEST

                                          ;;*****************************************************************
                                          ;*TEST 70                INVALID SECTOR/TRACK TEST

                                          ;;*****************************************************************
035556  000004                            TST70:  SCOPE
035560  012737  000070  001226                    MOV     #70,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

035566  000240                                    NOP
035570  012737  000024  001120                    MOV     #20.,$ICNT      ;20 ITERATIONS
035576  112737  000001  001131                    MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
035604  012737  035620  001122                    MOV     #T70,$LPADR  ;LOAD LOOP ON TEST ADDRESS
035612  012737  035620  001124                    MOV     #T70,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
035620                                    T70:
035620  012706  001100                            MOV     #STACK,SP       ;LOAD THE STACK POINTER
035624  013700  001276                            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
035630  013701  001456                            MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
035634  012702  036014                            MOV     #100$,R2                ;INITIALIZE TABLE POINTER

035640                                    10$:
                                          ;SET VOLUME VALID USING SUBROUTINE
035640  004737  060526                            JSR     PC,SETVV        ;GO SET VOLUME VALID
035644  000402                                    BR      20$             ;BRANCH TO 20$ IF NO ERROR
035646  104000                                    ERROR                   ;RETURN HERE IF ERROR
035650  000460                                    BR      50$                     ;SKIP REST OF TEST
035652                                    20$:
                                          ;CLEAR DESIRED CYLINDER, LOAD SECTOR/TRACK ADDRESS FROM TABLE, AND SET
                                          ;18 BIT FORMAT

035652  012760  000000  000034                    MOV     #0,RMDC(R0)     ;LOAD RMDC

035660  012760  000000  000032                    MOV     #0,RMOF(R0)     ;LOAD RMOF

035666  011260  000006                            MOV     (R2),RMDA(R0)   ;LOAD RMDA
                                          ;EXECUTE A SEARCH COMMAND TO WHERE "SET PULSE" IS ACTIVE
```

```
035672  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

035700  012760  000031  000000          MOV     #SEARCH!GO,RMCS1(R0)    ;LOAD RMCS1
035706  012703  000002                  MOV     #2,R3
035712                          30$:

035712  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

035720  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
035726  005303                          DEC     R3
035730  001370                          BNE     30$                     ;ISSUE 2 CLOCKS
                                ;VERIFY IAE IS SET

035732  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
035740  042737  175777  001142          BIC     #↑CIAE,$BDDAT
035746  001014                          BNE     40$                     ;BRANCH IF IAE IS ON
035750  012737  002000  001140          MOV     #IAE,$GDDAT             ;SETUP ERROR MESSAGE
035756  010037  001136                  MOV     R0,$BDADR
035762  062737  000014  001136          ADD     #RMER1,$BDADR
035770  011237  001174                  MOV     (R2),$TMP0
035774  104210                          ERROR   210                     ;IAE NOT SET BY RMDA ADDRESS
035776  000405                          BR      50$
036000                          40$:
                                ;ADVANCE TO NEXT ENTRY IN TABLE - EXIT IF DONE
036000  062702  000002                  ADD     #2,R2
036004  005712                          TST     (R2)
036006  001401                          BEQ     50$                     ;EXIT IF END OF TABLE
036010  000713                          BR      10$                     ;REPEAT TEST
036012  000414                  50$:    BR      200$                    ;JUMP OVER TABLE

036014                          100$:

                                ;TABLE OF SECTOR AND TRACK ADDRESSES FOR TEST
036014     000                          .BYTE   0                       ;SECTOR ADDRESS = 0
036015     005                          .BYTE   ↑B00000101              ;TRACK ADDRESS = 5

036016     000                          .BYTE   0                       ;SECTOR = 0
036017     006                          .BYTE   ↑B00000110              ;TRACK  = 6

036020     000                          .BYTE   0                       ;SECTOR = 0
036021     010                          .BYTE   ↑B00001000              ;TRACK = 8

036022     000                          .BYTE   0                       ;SECTOR = 0
036023     020                          .BYTE   ↑B00010000              ;TRACK = 16

036024     000                          .BYTE   0                       ;SECTOR = 0
036025     040                          .BYTE   ↑B00100000              ;TRACK = 32

036026     000                          .BYTE   0                       ;SECTOR = 0
036027     100                          .BYTE   ↑B01000000              ;TRACK = 64

036030     000                          .BYTE   0                       ;SECTOR = 0
036031     200                          .BYTE   ↑B10000000              ;TRACK = 128

036032     036                          .BYTE   ↑B00011110              ;SECTOR = 31
036033     000                          .BYTE   0                       ;TRACK = 0
```

```

036034     040                                   .BYTE   ↑B00100000                ;SECTOR = 32
036035     000                                   .BYTE   0                         ;TRACK = 0

036036     100                                   .BYTE   ↑B01000000                ;SECTOR = 64
036037     000                                   .BYTE   0                         ;TRACK = 0

036040     200                                   .BYTE   ↑B10000000                ;SECTOR = 128
036041     000                                   .BYTE   0                         ;TRACK = 0

036042     000                                   .BYTE   0                         ;END OF TABLE
036043     000                                   .BYTE   0

036044                                   200$:                                     ;END OF TEST


                                         ;;*****************************************************
                                         ;*TEST 71        INVALID CYLINDER TEST

                                         ;;*****************************************************
036044  000004                           TST71:  SCOPE
036046  012737  000071  001226                  MOV     #71,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX

036054  000240                                   NOP
036056  012737  000024  001120                  MOV     #20.,$ICNT        ;20 ITERATIONS
036064  112737  000001  001131                  MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
036072  012737  036106  001122                  MOV     #T71,$LPADR  ;LOAD LOOP ON TEST ADDRESS
036100  012737  036106  001124                  MOV     #T71,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
036106                                   T71:
036106  012706  001100                          MOV     #STACK,SP         ;LOAD THE STACK POINTER
036112  013700  001276                          MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
036116  013701  001456                          MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
036122  012702  036274                          MOV     #100$,R2                  ;INITIALIZE TABLE POINTER
036126                                   10$:
                                         ;SET VOLUME VALID USING SUBROUTINE
036126  004737  060526                          JSR     PC,SETVV          ;GO SET VOLUME VALID
036132  000402                                  BR      20$               ;BRANCH TO 20$ IF NO ERROR
036134  104000                                  ERROR                     ;RETURN HERE IF ERROR
036136  000455                                  BR      50$                       ;SKIP IF ERROR
036140                                   20$:
                                         ;CLEAR RMDA, LOAD RMDC FROM TABLE

036140  012760  000000  000006                  MOV     #0,RMDA(R0)       ;LOAD RMDA

036146  011260  000034                          MOV     (R2),RMDC(R0)     ;LOAD RMDC
                                         ;ENABLE DEBUG CLOCK AND EXECUTE SEARCH COMMAND

036152  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

036160  012760  000031  000000                  MOV     #SEARCH!GO,RMCS1(R0)     ;LOAD RMCS1
036166  012703  000002                          MOV     #2,R3
036172                                   30$:

036172  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1

036200  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
036206  005303                                  DEC     R3
```

```
036210  001370                          BNE     30$                     ;ISSUE 2 CLOCKS
                                ;VERIFY IAE IS SET

036212  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
036220  042737  175777  001142          BIC     #↑C<IAE>,$BDDAT
036226  001014                          BNE     40$                     ;BRANCH IF IAE IS SET
036230  012737  002000  001140          MOV     #IAE,$GDDAT             ;SETUP ERROR MESSAGE
036236  010037  001136                  MOV     R0,$BDADR
036242  062737  000014  001136          ADD     #RMER1,$BDADR
036250  011237  001174                  MOV     (R2),$TMP0
036254  104211                          ERROR   211                     ;IAE NOT SET BY RMDC
036256  000405                          BR      50$
036260                          40$:
                                ;ADVANCE TABBLE POINTER - EXIT IF DONE
036260  062702  000002                  ADD     #2,R2
036264  005712                          TST     (R2)
036266  001401                          BEQ     50$
036270  000716                          BR      10$
036272  000405                  50$:    BR      200$                    ;JUMP OVER TABLE

036274                          100$:

                                ;TABLE OF CYLINER ADDRESSES USED DURING TEST

036274  001467                          .WORD   ↑B1100110111            ;CYLINDER 823

036276  001470                          .WORD   ↑B1100111000            ;CYLINDER 824

036300  001500                          .WORD   ↑B1101000000            ;CYLINDER 832

036302  001600                          .WORD   ↑B1110000000            ;CYLINDER 896

036304  000000                          .WORD   0                       ;END OF TABLE

036306                          200$:                                   ;END OF TEST


                                ;;*****************************************************************
                                ;*TEST 72        SET AOE TEST

                                ;;*****************************************************************
036306  000004                  TST72:  SCOPE
036310  012737  000072  001226          MOV     #72,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

036316  000240                          NOP
036320  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
036326  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
036334  012737  036350  001122          MOV     #T72,$LPADR  ;LOAD LOOP ON TEST ADDRESS
036342  012737  036350  001124          MOV     #T72,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
036350                          T72:
036350  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
036354  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
036360  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                ;SET VOLUME VALID USING SUBROUTINE
036364  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
036370  000402                          BR      10$             ;BRANCH TO 10$ IF NO ERROR
036372  104000                          ERROR                   ;RETURN HERE IF ERROR
```

```
036374  000465                          BR      30$                         ;SKIP TEST IF ERROR
036376                          10$:
                                ;ENABLE DEBUG CLOCK AND LOAD LAST SECTOR ADDRESS, MEMORY ADDRESS AND
                                ;WORD COUNT, THEN LOAD WRITE DATA COMMAND WITH GO SET
036376  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
036404  012760  000000  000032          MOV     #0,RMOF(R0)      ;LOAD RMOF
036412  012760  002035  000006          MOV     #002035,RMDA(R0)            ;LOAD RMDA
036420  012760  001466  000034          MOV     #001466,RMDC(R0)            ;LOAD RMDC
036426  012760  106652  000004          MOV     #BUFFER,RMBA(R0)            ;LOAD RMBA
036434  012760  177244  000002          MOV     #↑C<2*256>+1,RMWC(R0)       ;LOAD RMWC
036442  012760  000061  000000          MOV     #WD!GO,RMCS1(R0)            ;LOAD RMCS1
036450  012702  000002                  MOV     #2,R2                       ;R2 = CLOCK COUNT
                                ;CLOCK THE COMMAND SEQUENCER TO GENERATE SET PULSE
036454                          20$:
036454  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1
036462  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
036470  005302                          DEC     R2
036472  001370                          BNE     20$                         ;ISSUE 2 CLOCKS
                                ;FORCE EBL TO GET TO OVERFLOW ADDRESS
036474  012760  061001  000024          MOV     #DMD!MUR!DBEN!DEBL,RMMR1(R0)     ;LOAD RMMR1
036502  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                ;VERIFY THAT ADDRESS OVERFLOW ERROR IS SET
036510  016037  000014  001142          MOV     RMER1(R0),$BDDAT            ;STORE RMER1 AT $BDDAT
036516  042737  176777  001142          BIC     #↑CAOE,$BDDAT
036524  001011                          BNE     30$                         ;BRANCH IF AOE IS SET
036526  010037  001136                  MOV     R0,$BDADR                   ;SETUP ERROR MESSAGE
036532  062737  000014  001136          ADD     #RMER1,$BDADR
036540  012737  001000  001140          MOV     #AOE,$GDDAT
036546  104212                          ERROR   212                         ;CANT SET AOE
036550                          30$:                                        ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 73        SET RMR TEST

                                ;;*****************************************************************
036550  000004                  TST73:  SCOPE
036552  012737  000073  001226          MOV     #73,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

036560  000240                          NOP
036562  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
036570  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
036576  012737  036612  001122          MOV     #T73,$LPADR ;LOAD LOOP ON TEST ADDRESS
036604  012737  036612  001124          MOV     #T73,$LPERR ;LOAD LOOP ON ERROR ADDRESS
036612                          T73:
```

```
036612  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
036616  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
036622  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
036626  010037  001136                  MOV     R0,$BDADR                       ;SETUP REGISTER ADDRESS FOR MSG
036632  062737  000014  001136          ADD     #RMER1,$BDADR
036640  012702  037014                  MOV     #100$,R2                        ;INITIALIZE TABLE POINTER
036644                          10$:
                                ;CLEAR THE DEVICE AND ENABLE DEBUG CLOCK
036644  012760  000040  000010          MOV     #CLR,RMCS2(R0)          ;CLEAR THE MASSBUS
036652  111160  000010                  MOVB    (R1),RMCS2(R0)          ;SELECT UNIT

036656  012760  000001  000024          MOV     #DMD,RMMR1(R0)          ;LOAD RMMR1

036664  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

036672  012760  000000  000014          MOV     #0,RMER1(R0)            ;LOAD RMER1

036700  012760  000000  000042          MOV     #0,RMER2(R0)            ;LOAD RMER2
                                ;SET GO THEN WRITE THE REGISTER SPECIFIED BY THE TABLE

036706  012760  000001  000000          MOV     #GO,RMCS1(R0)           ;LOAD RMCS1
036714  011203                          MOV     (R2),R3                         ;GENERATE REGISTER ADDRESS
036716  060003                          ADD     R0,R3
036720  012713  041001                  MOV     #DMD!MUR!DBEN,(R3)              ;WRITE THE REGISTER
                                ;VERIFY RMR ACCORDING TO TABLE

036724  016037  000014  001142          MOV     RMER1(R0),$BDDAT                ;STORE RMER1 AT $BDDAT
036732  042737  177773  001142          BIC     #↑CRMR,$BDDAT
036740  016237  000002  001140          MOV     2(R2),$GDDAT                    ;GET EXPECTED RESULT FROM TABLE
036746  023737  001140  001142          CMP     $GDDAT,$BDDAT
036754  001411                          BEQ     30$                             ;BRANCH IF RMR IS OK
036756  011237  001174                  MOV     (R2),$TMP0                      ;SAVE TEST REGISTER
036762  032762  000004  000002          BIT     #RMR,2(R2)
036770  001002                          BNE     20$                             ;BRANCH IF ERROR SHOULD BE ONE
036772  104213                          ERROR   213                             ;RMR SET WHEN WRITING REGISTER
036774  000401                          BR      30$
036776  104214                  20$:    ERROR   214                             ;RMR DID NOT SET

037000                          30$:
                                ;ADVANCE TABLE POINTER, EXIT IF DONE
037000  062702  000004                  ADD     #4,R2
037004  005712                          TST     (R2)
037006  100401                          BMI     40$                             ;EXIT IF ENTRY NEGATIVE
037010  000715                          BR      10$
037012  000410                  40$:    BR      200$                            ;JUMP OVER TABLE

037014                          100$:

                                ;TABLE OF REGISTER ADDRESSES AND RMR VALUES

037014  000016                          .WORD   #RMAS                           ;ATTENTION SUMMARY REG
037016  000000                          .WORD                                   ;RMR = 0

037020  000024                          .WORD   #RMMR1                          ;MAINTENANCE REG
037022  000000                          .WORD                                   ;RMR = 0
```

```
037024  000006                          .WORD   #RMDA                           ;DISK ADDRESS REG
037026  000004                          .WORD   RMR                             ;RMR = 1
037030  177777                          .WORD   #-1                             ;END OF TABLE
037032  000000                          .WORD
037034                          200$:                                           ;END OF TEST
                                ;;*****************************************************************
                                ;*TEST 74        PGM STATUS CHECK
                                ;;*****************************************************************
037034  000004                  TST74:  SCOPE
037036  012737  000074  001226          MOV     #74,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX
037044  000240                          NOP
037046  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
037054  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
037062  012737  037076  001122          MOV     #T74,$LPADR     ;LOAD LOOP ON TEST ADDRESS
037070  012737  037076  001124          MOV     #T74,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
037076                          T74:
037076  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
037102  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
037106  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
                                ;CLEAR AND READ DRIVE TYPE AND DRIVE STATUS
037112  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
037120  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
037124  016037  000026  001174          MOV     RMDT(R0),$TMP0  ;STORE RMDT AT $TMP0
037132  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
                                ;OMIT TEST IF DRQ IS ON - ELSE VERIFY THAT PGM IS OFF
037140  032737  004000  001174          BIT     #DRQ,$TMP0
037146  001014                          BNE     10$                             ;BRANCH IF DRQ IS ON
037150  042737  176777  001142          BIC     #↑CPGM,$BDDAT
037156  001410                          BEQ     10$                             ;BRANCH IF PGM IS OFF
037160  005037  001140                  CLR     $GDDAT
037164  010037  001136                  MOV     R0,$BDADR
037170  062737  000012  001134          ADD     #RMDS,$GDADR
037176  104215                          ERROR   215                             ;DRQ IS OFF BUT PGM IS ON
037200                          10$:                                            ;END OF TEST
                                ;;*****************************************************************
                                ;*TEST 75        DPR STATUS CHECK
                                ;;*****************************************************************
037200  000004                  TST75:  SCOPE
037202  012737  000075  001226          MOV     #75,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX
037210  000240                          NOP
037212  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
037220  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
037226  012737  037242  001122          MOV     #T75,$LPADR     ;LOAD LOOP ON TEST ADDRESS
037234  012737  037242  001124          MOV     #T75,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
037242                          T75:
037242  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
```

```
037246  013700  001276                  MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
037252  013701  001456                  MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
                                ;CLEAR AND VERIFY THAT DVA IS SET
037256  012760  000040  000010          MOV     #CLR,RMCS2(R0)    ;CLEAR THE MASSBUS
037264  111160  000010                  MOVB    (R1),RMCS2(R0)    ;SELECT UNIT

037270  016037  000000  001142          MOV     RMCS1(R0),$BDDAT          ;STORE RMCS1 AT $BDDAT
037276  042737  173777  001142          BIC     #↑CDVA,$BDDAT
037304  001006                          BNE     10$                       ;BRANCH IF DVA IS ON
037306  012737  004000  001140          MOV     #DVA,$GDDAT
037314  010037  001136                  MOV     R0,$BDADR
037320  104216                          ERROR   216                       ;DVA NOT SET
037322                          10$:
                                ;VERIFY THAT DPR IS SET

037322  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
037330  042737  177377  001142          BIC     #↑CDPR,$BDDAT
037336  001011                          BNE     20$
037340  012737  000400  001140          MOV     #DPR,$GDDAT
037346  010037  001136                  MOV     R0,$BDADR
037352  062737  000012  001136          ADD     #RMDS,$BDADR
037360  104217                          ERROR   217                       ;DPR NOT SET

037362                          20$:                                      ;END OF TEST

                                ;;*************************************************************
                                ;*TEST 76        PORT REQUEST TEST, PART 1

                                ;;*************************************************************
037362  000004                  TST76:  SCOPE
037364  012737  000076  001226          MOV     #76,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX

037372  000240                          NOP
037374  012737  000024  001120          MOV     #20.,$ICNT        ;20 ITERATIONS
037402  112737  000001  001131          MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
037410  012737  037424  001122          MOV     #T76,$LPADR  ;LOAD LOOP ON TEST ADDRESS
037416  012737  037424  001124          MOV     #T76,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
037424                          T76:
037424  012706  001100                  MOV     #STACK,SP         ;LOAD THE STACK POINTER
037430  013700  001276                  MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
037434  013701  001456                  MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
                                ;SET VOLUME VALID USING SUBROUTINE
037440  004737  060526                  JSR     PC,SETVV          ;GO SET VOLUME VALID
037444  000402                          BR      10$               ;BRANCH TO 10$ IF NO ERROR
037446  104000                          ERROR                     ;RETURN HERE IF ERROR
037450  000434                          BR      20$
037452                          10$:
                                ;EXECUTE A RELEASE TO RESET REQUEST FLOP

037452  012760  000013  000000          MOV     #RLEASE!GO,RMCS1(R0)      ;LOAD RMCS1
                                ;READ RMMR2 AND SKIP TEST IF REQUEST FLOPS ARENT RESET

037460  016002  000040                  MOV     RMMR2(R0),R2      ;STORE RMMR2 AT R2
037464  042702  037777                  BIC     #↑C<RQA!RQB>,R2
037470  001024                          BNE     20$
                                ;READ RMCS1 TO SET REQUEST FLOP
```

```

037472  016002  000000                          MOV     RMCS1(R0),R2      ;STORE RMCS1 AT R2
                                        ;VERIFY THAT REQUEST FLOP IS SET IF PGM IS SET

037476  016005  000012                          MOV     RMDS(R0),R5       ;STORE RMDS AT R5
037502  032705  001000                          BIT     #PGM,R5           ;SEE IF PGM IS SET
037506  001415                                  BEQ     20$               ;DONT TEST REQUEST IF PGM IS ZERO

037510  016037  000040  001142                  MOV     RMMR2(R0),$BDDAT          ;STORE RMMR2 AT $BDDAT
037516  042737  037777  001142                  BIC     #↑C<RQA!RQB>,$BDDAT
037524  001006                                  BNE     20$                       ;BRANCH IF REQUEST IS SET
037526  010037  001136                          MOV     R0,$BDADR
037532  062737  000040  001136                  ADD     #RMMR2,$BDADR
037540  104220                                  ERROR   220                       ;CANT SET REQUEST BY RMCS1
037542                                  20$:

                                        ;;*************************************************************
                                        ;*TEST 77           PORT REQUEST TEST, PART 2

                                        ;;*************************************************************
037542  000004                          TST77:  SCOPE
037544  012737  000077  001226                  MOV     #77,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX

037552  000240                                  NOP
037554  012737  000024  001120                  MOV     #20.,$ICNT        ;20 ITERATIONS
037562  112737  000001  001131                  MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
037570  012737  037604  001122                  MOV     #T77,$LPADR  ;LOAD LOOP ON TEST ADDRESS
037576  012737  037604  001124                  MOV     #T77,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
037604                                  T77:
037604  012706  001100                          MOV     #STACK,SP         ;LOAD THE STACK POINTER
037610  013700  001276                          MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
037614  013701  001456                          MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
                                        ;SET VOLUME VALID USING SUBROUTINE
037620  004737  060526                          JSR     PC,SETVV          ;GO SET VOLUME VALID
037624  000402                                  BR      10$               ;BRANCH TO 10$ IF NO ERROR
037626  104000                                  ERROR                     ;RETURN HERE IF ERROR
037630  000435                                  BR      20$
037632                                  10$:
                                        ;EXECUTE A RELEASE TO RESET REQUEST FLOP

037632  012760  000013  000000                  MOV     #RLEASE!GO,RMCS1(R0)      ;LOAD RMCS1
                                        ;READ RMMR2 AND SKIP TEST IF REQUEST FLOPS ARENT RESET

037640  016002  000040                          MOV     RMMR2(R0),R2      ;STORE RMMR2 AT R2
037644  042702  037777                          BIC     #↑C<RQA!RQB>,R2
037650  001025                                  BNE     20$
                                        ;WRITE THE ATTENTION SUMMARY REGISTER TO SET REQUEST FLOP

037652  012760  177777  000016                  MOV     #-1,RMAS(R0)      ;LOAD RMAS
                                        ;VERIFY THAT REQUEST FLOP IS SET IF PGM IS SET

037660  016005  000012                          MOV     RMDS(R0),R5       ;STORE RMDS AT R5
037664  032705  001000                          BIT     #PGM,R5           ;SEE IF PGM IS SET
037670  001415                                  BEQ     20$               ;DONT TEST REQUEST IF PGM IS ZERO

037672  016037  000040  001142                  MOV     RMMR2(R0),$BDDAT          ;STORE RMMR2 AT $BDDAT
```

```
037700  042737  037777  001142          BIC     #↑C<RQA!RQB>,$BDDAT
037706  001006                          BNE     20$
037710  010037  001136                  MOV     R0,$BDADR
037714  062737  000040  001136          ADD     #RMMR2,$BDADR
037722  104221                          ERROR   221                             ;CANT SEE REQUEST BY RMAS
037724                          20$:
                                ;;*****************************************************************
                                ;*TEST 100       PORT REQUEST TEST, PART 3
                                ;;*****************************************************************
037724  000004                  TST100: SCOPE
037726  012737  000100  001226          MOV     #100,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
037734  000240                          NOP
037736  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
037744  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
037752  012737  037766  001122          MOV     #T100,$LPADR  ;LOAD LOOP ON TEST ADDRESS
037760  012737  037766  001124          MOV     #T100,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
037766                          T100:
037766  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
037772  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
037776  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                ;SET VOLUME VALID USING SUBROUTINE
040002  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
040006  000402                          BR      10$             ;BRANCH TO 10$ IF NO ERROR
040010  104000                          ERROR                   ;RETURN HERE IF ERROR
040012  000435                          BR      20$
040014                          10$:
                                ;EXECUTE A RELEASE COMMAND TO RESET REQUEST FLOP
040014  012760  000013  000000          MOV     #RLEASE!GO,RMCS1(R0)    ;LOAD RMCS1
                                ;READ RMMR2 AND SKIP TEST IF REQUEST FLOPS ARENT RESET
040022  016002  000040                  MOV     RMMR2(R0),R2    ;STORE RMMR2 AT R2
040026  042702  037777                  BIC     #↑C<RQA!RQB>,R2
040032  001025                          BNE     20$
                                ;WRITE RMDA TO SET REQUEST FLOP
040034  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA
                                ;VERIFY THAT REQUEST FLOP IS SET IF PGM IS SET
040042  016005  000012                  MOV     RMDS(R0),R5     ;STORE RMDS AT R5
040046  032705  001000                  BIT     #PGM,R5         ;SEE IF PGM IS SET
040052  001415                          BEQ     20$             ;DONT TEST REQUEST IF PGM IS ZERO
040054  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
040062  042737  037777  001142          BIC     #↑C<RQA!RQB>,$BDDAT
040070  001006                          BNE     20$
040072  010037  001136                  MOV     R0,$BDADR
040076  062737  000040  001136          ADD     #RMMR2,$BDADR
040104  104222                          ERROR   222                     ;CANT SET REQUEST BY RMDA
040106                          20$:
                                ;;*****************************************************************
```

```
                                        ;*TEST 101       RELEASE TEST

                                        ;;*****************************************************
040106  000004                          TST101: SCOPE
040110  012737  000101  001226                  MOV     #101,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

040116  000240                                  NOP
040120  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
040126  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
040134  012737  040150  001122                  MOV     #T101,$LPADR  ;LOAD LOOP ON TEST ADDRESS
040142  012737  040150  001124                  MOV     #T101,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
040150                                  T101:
040150  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
040154  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
040160  013701  001456                          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                        ;REQUEST FLOP SHOULD SET WHEN WRITING RMCS1

                                        ;SET VOLUME VALID USING SUBROUTINE
040164  004737  060526                          JSR     PC,SETVV        ;GO SET VOLUME VALID
040170  000402                                  BR      10$             ;BRANCH TO 10$ IF NO ERROR
040172  104000                                  ERROR                   ;RETURN HERE IF ERROR
040174  000454                                  BR      40$
040176                                  10$:
                                        ;EXECUTE A RELEASE COMMAND

040176  012760  041001  000024                  MOV     #DMD!DBEN!MUR,RMMR1(R0) ;LOAD RMMR1

040204  012760  000013  000000                  MOV     #RLEASE!GO,RMCS1(R0)    ;LOAD RMCS1
040212  012702  000002                          MOV     #2,R2
040216                                  20$:

040216  012760  141001  000024                  MOV     #DMD!DBEN!MUR!DBCK,RMMR1(R0)    ;LOAD RMMR1

040224  012760  041001  000024                  MOV     #DMD!DBEN!MUR,RMMR1(R0) ;LOAD RMMR1
040232  005302                                  DEC     R2
040234  001370                                  BNE     20$                     ;ISSUE 2 CLOCKS
                                        ;VERIFY REQUEST FLOPS ARE RESET

040236  016037  000026  001174                  MOV     RMDT(R0),$TMP0  ;STORE RMDT AT $TMP0

040244  016037  000040  001142                  MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
040252  042737  037777  001142                  BIC     #↑C<RQA!RQB>,$BDDAT
040260  001422                                  BEQ     40$                     ;BRANCH IF REQUESTS ARE RESET
040262  032737  004000  001174                  BIT     #DRQ,$TMP0
040270  001410                                  BEQ     30$                     ;BRANCH IF SINGLE PORT DEVICE
040272  032737  100000  001142                  BIT     #RQA,$BDDAT
040300  001412                                  BEQ     40$                     ;BRANCH IF RQA IS RESET
040302  032737  040000  001142                  BIT     #RQB,$BDDAT
040310  001406                                  BEQ     40$                     ;BRANCH IF RQB IS RESET
040312                                  30$:
                                        ;DRQ IS ZERO AND A REQUEST FLOP IS ON, OR, DRQ IS ONE AND
                                        ;BOTH REQUEST FLOPS ARE ON
040312  010037  001136                          MOV     R0,$BDADR
040316  062737  000040  001136                  ADD     #RMMR2,$BDADR
040324  104223                                  ERROR   223                     ;REQUEST FLOP NOT RESET
040326                                  40$:                                    ;END OF TEST
```

```
                                          ;;*************************************************************
                                          ;*TEST 102        WRITE ATA TEST
                                          ;;*************************************************************
040326  000004                            TST102: SCOPE
040330  012737  000102  001226                    MOV     #102,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
040336  000240                                    NOP
040340  012737  000024  001120                    MOV     #20.,$ICNT      ;20 ITERATIONS
040346  112737  000001  001131                    MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
040354  012737  040370  001122                    MOV     #T102,$LPADR    ;LOAD LOOP ON TEST ADDRESS
040362  012737  040370  001124                    MOV     #T102,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
040370                                    T102:
040370  012706  001100                            MOV     #STACK,SP       ;LOAD THE STACK POINTER
040374  013700  001276                            MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
040400  013701  001456                            MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                          ;CLEAR THE DEVICE, SET DIAGNOSTIC MODE THEN SET UNIT READY
040404  012760  000040  000010                    MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
040412  111160  000010                            MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
040416  012760  000001  000024                    MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
040424  012760  000000  000014                    MOV     #0,RMER1(R0)    ;LOAD RMER1
040432  012760  000000  000042                    MOV     #0,RMER2(R0)    ;LOAD RMER2
040440  012760  001001  000024                    MOV     #DMD!MUR,RMMR1(R0)              ;LOAD RMMR1
                                          ;WRITE THE ATTENTION SUMMARY REGISTER
040446  111102                                    MOVB    (R1),R2                         ;ASSEMBLE THE ATA BIT IN R3
040450  042702  177770                            BIC     #↑CUNTMSK,R2
040454  116203  066064                            MOVB    ATNTBL(R2),R3
040460  042703  177400                            BIC     #↑CATNMSK,R3
040464  010360  000016                            MOV     R3,RMAS(R0)     ;LOAD RMAS
                                          ;READ RMDS AND VERIFY THAT ATA IS RESET
040470  016037  000012  001142                    MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
040476  042737  077777  001142                    BIC     #↑CATA,$BDDAT
040504  001411                                    BEQ     10$                             ;BRANCH IF ATA IS RESET
040506  010037  001136                            MOV     R0,$BDADR
040512  062737  000012  001136                    ADD     #RMDS,$BDADR
040520  005037  001140                            CLR     $GDDAT
040524  104224                                    ERROR   224                             ;CANT CLEAR ATA
040526  000417                                    BR      20$
040530                                    10$:
                                          ;READ RMAS AND VERIFY ATA IS RESET
040530  016037  000016  001142                    MOV     RMAS(R0),$BDDAT ;STORE RMAS AT $BDDAT
040536  005103                                    COM     R3
040540  040337  001142                            BIC     R3,$BDDAT
040544  001410                                    BEQ     20$                             ;BRANCH IF ATA IS RESET
040546  005037  001140                            CLR     $GDDAT
040552  010037  001136                            MOV     R0,$BDADR
040556  062737  000016  001136                    ADD     #RMAS,$BDADR
040564  104225                                    ERROR   225                             ;RMAS ATA NOT ZERO
040566                                    20$:                                            ;END OF TEST
```

```
                                      ;;***********************************************************
                                      ;*TEST 103        RESET ATA BY GO TEST

                                      ;;***********************************************************
040566  000004                        TST103: SCOPE
040570  012737  000103  001226                MOV     #103,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

040576  000240                                NOP
040600  012737  000024  001120                MOV     #20.,$ICNT      ;20 ITERATIONS
040606  112737  000001  001131                MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
040614  012737  040630  001122                MOV     #T103,$LPADR    ;LOAD LOOP ON TEST ADDRESS
040622  012737  040630  001124                MOV     #T103,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
040630                                T103:
040630  012706  001100                        MOV     #STACK,SP       ;LOAD THE STACK POINTER
040634  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
040640  013701  001456                        MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
040644  012760  000040  000010                MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
040652  111160  000010                        MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

040656  012760  000001  000024                MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

040664  012760  000000  000014                MOV     #0,RMER1(R0)    ;LOAD RMER1

040672  012760  000000  000042                MOV     #0,RMER2(R0)    ;LOAD RMER2

040700  012760  041001  000024                MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                      ;WITH DEBUG CLOCK ENABLED, SET GO AND VERIFY THAT ATA IS RESET

040706  012760  000001  000000                MOV     #GO,RMCS1(R0)   ;LOAD RMCS1

040714  016037  000012  001142                MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
040722  042737  077777  001142                BIC     #↑CATA,$BDDAT
040730  001410                                BEQ     10$                     ;BRANCH IF ATA IS RESET
040732  005037  001140                        CLR     $GDDAT
040736  010037  001136                        MOV     R0,$BDADR
040742  062737  000012  001136                ADD     #RMDS,$BDADR
040750  104226                                ERROR   226                     ;CANT RESET ATA WITH GO

040752                                10$:                                    ;END OF TEST

                                      ;;***********************************************************
                                      ;*TEST 104        UNIT READY ATA TEST

                                      ;;***********************************************************
040752  000004                        TST104: SCOPE
040754  012737  000104  001226                MOV     #104,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

040762  000240                                NOP
040764  012737  000024  001120                MOV     #20.,$ICNT      ;20 ITERATIONS
040772  112737  000001  001131                MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
041000  012737  041014  001122                MOV     #T104,$LPADR    ;LOAD LOOP ON TEST ADDRESS
041006  012737  041014  001124                MOV     #T104,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
041014                                T104:
041014  012706  001100                        MOV     #STACK,SP       ;LOAD THE STACK POINTER
041020  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
```

```
041024  013701  001456                    MOV     TSTQUE,R1          ;R1 = POINTER TO DEVICE
                                  ;SET DIAGNOSTIC MODE AND CLEAR ATA
041030  012760  000040  000010            MOV     #CLR,RMCS2(R0)     ;CLEAR THE MASSBUS
041036  111160  000010                    MOVB    (R1),RMCS2(R0)     ;SELECT UNIT
041042  012760  000001  000024            MOV     #DMD,RMMR1(R0)     ;LOAD RMMR1
041050  012760  000000  000014            MOV     #0,RMER1(R0)       ;LOAD RMER1
041056  012760  000000  000042            MOV     #0,RMER2(R0)       ;LOAD RMER2
041064  012760  177777  000016            MOV     #-1,RMAS(R0)       ;LOAD RMAS
                                  ;SET UNIT READY AND VERIFY ATA IS SET
041072  012760  001001  000024            MOV     #DMD!MUR,RMMR1(R0)          ;LOAD RMMR1
041100  016037  000012  001142            MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
041106  042737  077777  001142            BIC     #^CATA,$BDDAT
041114  001011                            BNE     10$                         ;BRANCH IF ATA IS SET
041116  010037  001136                    MOV     R0,$BDADR
041122  062737  000012  001136            ADD     #RMDS,$BDADR
041130  012737  100000  001140            MOV     #ATA,$GDDAT
041136  104227                            ERROR   227                         ;ATA NOT SET BY 01 READY
041140                            10$:
                                  ;CLEAR ATA, RESET UNIT READY, AND VERIFY ATA IS SET
041140  012760  177777  000016            MOV     #-1,RMAS(R0)       ;LOAD RMAS
041146  012760  000001  000024            MOV     #DMD,RMMR1(R0)     ;LOAD RMMR1
041154  016037  000012  001142            MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
041162  042737  077777  001142            BIC     #^CATA,$BDDAT
041170  001011                            BNE     20$
041172  010037  001136                    MOV     R0,$BDADR
041176  062737  000012  001136            ADD     #RMDS,$BDADR
041204  012737  100000  001140            MOV     #ATA,$GDDAT
041212  104230                            ERROR   230                         ;ATA NOT SET BY 10 READY
041214                            20$:                                        ;END OF TEST

                                  ;;*****************************************************************
                                  ;*TEST 105        ERROR ATA TEST

                                  ;;*****************************************************************
041214  000004                    TST105: SCOPE
041216  012737  000105  001226            MOV     #105,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
041224  000240                            NOP
041226  012737  000024  001120            MOV     #20.,$ICNT         ;20 ITERATIONS
041234  112737  000001  001131            MOVB    #1,$ERMAX          ;ONE ERROR ALLOWED
041242  012737  041256  001122            MOV     #T105,$LPADR  ;LOAD LOOP ON TEST ADDRESS
041250  012737  041256  001124            MOV     #T105,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
041256                            T105:
041256  012706  001100                    MOV     #STACK,SP          ;LOAD THE STACK POINTER
041262  013700  001276                    MOV     $BASE,R0           ;R0 = UNIBUS ADDRESS OF UUT
041266  013701  001456                    MOV     TSTQUE,R1          ;R1 = POINTER TO DEVICE
```

```
                                    ;CLEAR THE DEVICE AND RESET ATTENTION
041272  012760  000040  000010              MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
041300  111160  000010                      MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

041304  012760  000001  000024              MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

041312  012760  000000  000014              MOV     #0,RMER1(R0)    ;LOAD RMER1

041320  012760  000000  000042              MOV     #0,RMER2(R0)    ;LOAD RMER2

041326  012760  177777  000016              MOV     #-1,RMAS(R0)    ;LOAD RMAS
                                    ;WRITE ONES IN ERROR REGISTER 1 AND VERIFY ATA IS SET

041334  012760  177777  000014              MOV     #-1,RMER1(R0)   ;LOAD RMER1

041342  016037  000012  001142              MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
041350  042737  077777  001142              BIC     #↑CATA,$BDDAT
041356  001011                              BNE     10$
041360  012737  100000  001140              MOV     #ATA,$GDDAT
041366  010037  001136                      MOV     R0,$BDADR
041372  062737  000012  001136              ADD     #RMDS,$BDADR
041400  104231                              ERROR   231                     ;ATA NOT SET BY COMP ERROR

041402                              10$:                                    ;END OF TEST


                                    ;;*****************************************************************
                                    ;*TEST 106          REGISTER TRANSFER ATA TEST

                                    ;;*****************************************************************
041402  000004                      TST106: SCOPE
041404  012737  000106  001226              MOV     #106,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
041412  000240                              NOP
041414  012737  000024  001120              MOV     #20.,$ICNT      ;20 ITERATIONS
041422  112737  000001  001131              MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
041430  012737  041444  001122              MOV     #T106,$LPADR  ;LOAD LOOP ON TEST ADDRESS
041436  012737  041444  001124              MOV     #T106,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
041444                              T106:
041444  012706  001100                      MOV     #STACK,SP       ;LOAD THE STACK POINTER
041450  013700  001276                      MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
041454  013701  001456                      MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
041460  012702  041614                      MOV     #100$,R2                ;INITIALIZE TABLE ADDRESS
                                    ;FORCE COMPOSITE ERROR AND RESET ATA
041464                              5$:
041464  012760  000040  000010              MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
041472  111160  000010                      MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

041476  012760  000001  000024              MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

041504  012760  177777  000014              MOV     #-1,RMER1(R0)   ;LOAD RMER1

041512  012760  177777  000016              MOV     #-1,RMAS(R0)    ;LOAD RMAS
                                    ;WRITE THE TEST REGISTER AND VERIFY ATA
041520  011203                              MOV     (R2),R3                 ;GENERATE REGISTER ADDRESS
041522  060003                              ADD     R0,R3
041524  005013                              CLR     (R3)
```

```
041526 016037 000012 001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
041534 042737 077777 001142          BIC     #^CATA,$BDDAT
041542 016237 000002 001140          MOV     2(R2),$GDDAT
041550 023737 001140 001142          CMP     $GDDAT,$BDDAT
041556 001410                        BEQ     10$                     ;BRANCH IF ATA IS OK
041560 010337 001174                 MOV     R3,$TMP0
041564 010037 001136                 MOV     R0,$BDADR
041570 062737 000012 001136          ADD     #RMDS,$BDADR
041576 104232                        ERROR   232                     ;ATA INCORRECT FOR REG WRITTEN
041600                        10$:
                              ;MOVE TABLE POINTER - EXIT IF DONE
041600 062702 000004                 ADD     #4,R2
041604 005712                        TST     (R2)
041606 100401                        BMI     20$                     ;EXIT IF ENTRY MINUS
041610 000725                        BR      5$
041612 000410                 20$:   BR      200$                    ;JUMP OVER TABLE
041614                        100$:
                              ;TABLE OF REGISTER ADDRESSES AND ATA BITS
041614 000016                        .WORD   RMAS
041616 000000                        .WORD
041620 000006                        .WORD   RMDA
041622 100000                        .WORD   ATA
041624 000000                        .WORD   RMCS1
041626 000000                        .WORD
041630 177777                        .WORD   -1                      ;END OF TABLE
041632 000000                        .WORD
041634                        200$:                                  ;END OF TEST
                              ;;*****************************************************************
                              ;*TEST 107       P SET ATA TEST
                              ;;*****************************************************************
041634 000004                 TST107: SCOPE
041636 012737 000107 001226          MOV     #107,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
041644 000240                        NOP
041646 012737 000024 001120          MOV     #20.,$ICNT      ;20 ITERATIONS
041654 112737 000001 001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
041662 012737 041676 001122          MOV     #T107,$LPADR    ;LOAD LOOP ON TEST ADDRESS
041670 012737 041676 001124          MOV     #T107,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
041676                        T107:
041676 012706 001100                 MOV     #STACK,SP       ;LOAD THE STACK POINTER
041702 013700 001276                 MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
041706 013701 001456                 MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
041712 012702 042054                 MOV     #100$,R2        ;INITIALIZE TABLE POINTER
041716                        10$:
                              ;SET VOLUME VALID USING SUBROUTINE
041716 004737 060526                 JSR     PC,SETVV        ;GO SET VOLUME VALID
041722 000402                        BR      20$             ;BRANCH TO 20$ IF NO ERROR
```

```
041724  104000                                  ERROR                       ;RETURN HERE IF ERROR
041726  000451                                  BR      50$
041730                                  20$:
                                        ;EXECUTE THE COMMAND FROM THE TABLE

041730  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
041736  011203                                  MOV     (R2),R3             ;GET FUNCTION CODE
041740  052703  000001                          BIS     #GO,R3

041744  010360  000000                          MOV     R3,RMCS1(R0)        ;LOAD RMCS1
041750  012703  000003                          MOV     #3,R3
041754                                  30$:

041754  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1

041762  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
041770  005303                                  DEC     R3
041772  001370                                  BNE     30$
                                        ;VERIFY THAT ATA IS SET

041774  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
042002  042737  077777  001142                  BIC     #↑CATA,$BDDAT
042010  001013                                  BNE     40$
042012  010037  001136                          MOV     R0,$BDADR
042016  062737  000012  001136                  ADD     #RMDS,$BDADR
042024  012737  100000  001140                  MOV     #ATA,$GDDAT
042032  011237  001174                          MOV     (R2),$TMP0
042036  104233                                  ERROR   233                 ;ATA NOT SET BY SEQUENCER
042040                                  40$:
                                        ;ADVANCE TABLE POINTER-EXIT IF DONE
042040  062702  000002                          ADD     #2,R2
042044  005712                                  TST     (R2)
042046  100401                                  BMI     50$
042050  000722                                  BR      10$
042052  000403                          50$:    BR      200$                ;JUMP OVER TABLE

042054                                  100$:
                                        ;TABLE OF FUNCTION CODES

042054  000014                                  .WORD   OFFSET

042056  000016                                  .WORD   RTC

042060  177777                                  .WORD   -1                  ;END OF TABLE
042062                                  200$:

                                        ;;*****************************************************************
                                        ;*TEST 110          SET WLE TEST

                                        ;;*****************************************************************
042062  000004                          TST110: SCOPE
042064  012737  000110  001226                  MOV     #110,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX

042072  000240                                  NOP
042074  012737  000024  001120                  MOV     #20.,$ICNT          ;20 ITERATIONS
042102  112737  000001  001131                  MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
```

```
10811 042110 012737 042124 001122           MOV     #T110,$LPADR    ;LOAD LOOP ON TEST ADDRESS
10812 042116 012737 042124 001124           MOV     #T110,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
10813 042124                        T110:
10814 042124 012706 001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
10815 042130 013700 001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
10816 042134 013701 001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
10817 042140 012702 042334                  MOV     #100$,R2        ;INITIALIZE TABLE POINTER
10818 042144                        10$:
10819                               ;SET VOLUME VALID USING SUBROUTINE
10820 042144 004737 060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
10821 042150 000402                         BR      20$             ;BRANCH TO 20$ IF NO ERROR
10822 042152 104000                         ERROR                   ;RETURN HERE IF ERROR
10823 042154 000466                         BR      50$
10824 042156                        20$:
10825                               ;ENABLE DEBUG CLOCK AND SET WRITE PROTECT ACCORDING TO TABLE
10826 042156 016203 000002                  MOV     2(R2),R3        ;GET WRITE PROTECT FROM TABLE
10827 042162 052703 041001                  BIS     #DMD!MUR!DBEN,R3        ;SET OTHER MAINT BITS
10828
10829 042166 010360 000024                  MOV     R3,RMMR1(R0)    ;LOAD RMMR1
10830                               ;LOAD AND EXECUTE THE COMMAND FROM THE TABLE
10831 042172 011204                         MOV     (R2),R4         ;GET FUNCTION CODE FROM TABLE
10832 042174 052704 000001                  BIS     #GO,R4          ;SET GO
10833
10834 042200 010460 000000                  MOV     R4,RMCS1(R0)    ;LOAD RMCS1
10835 042204 012705 000002                  MOV     #2,R5           ;R5=CLOCK COUNT
10836 042210 052703 100000          30$:    BIS     #DBCK,R3        ;SET CLOCK
10837
10838 042214 010360 000024                  MOV     R3,RMMR1(R0)    ;LOAD RMMR1
10839 042220 042703 100000                  BIC     #DBCK,R3        ;RESET CLOCK
10840
10841 042224 010360 000024                  MOV     R3,RMMR1(R0)    ;LOAD RMMR1
10842 042230 005305                         DEC     R5
10843 042232 001366                         BNE     30$             ;ISSUE 2 CLOCKS
10844                               ;VERIFY THAT WRITE LOCK ERROR IS ACCORDING TO TABLE
10845
10846 042234 016037 000014 001142           MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
10847 042242 042737 173777 001142           BIC     #↑CWLE,$BDDAT
10848 042250 026237 000004 001142           CMP     4(R2),$BDDAT
10849 042256 001417                         BEQ     40$             ;BRANCH IF WLE IS OK
10850 042260 016237 000004 001140           MOV     4(R2),$GDDAT    ;SAVE DATA FOR ERROR MSG
10851 042266 010037 001136                  MOV     R0,$BDADR
10852 042272 062737 000014 001136           ADD     #RMER1,$BDADR
```

```
042300  011237  001174                    MOV     (R2),$TMPO          ;$TMPO=FUNCTION CODE
042304  016237  000002  001176            MOV     2(R2),$TMP1         ;$TMP1=WRITE PROTECT
042312  104234                            ERROR   234                 ;WLE INCORRECT
042314  000406                            BR      50$
042316                            40$:
                                  ;ADVANCE TABLE POINTER TO NEXT FUNCTION CODE-EXIT IF DONE
042316  062702  000006                    ADD     #6,R2
042322  005762  000002                    TST     2(R2)
042326  100401                            BMI     50$
042330  000705                            BR      10$                 ;REPEAT TEST

042332  000416                    50$:    BR      200$                ;JUMP OVER TABLE

042334                            100$:

                                  ;TABLE OF FUNCTION CODES, WRITE PROTECT AND WRITE LOCK ERRORS

042334  000060                            .WORD   WD                  ;WRITE DATA COMMAND
042336  000010                            .WORD   MWP                 ;WRITE PROTECT ON
042340  004000                            .WORD   WLE                 ;WRITE LOCK ERROR ONE

042342  000060                            .WORD   WD                  ;WRITE DATA COMMAND
042344  000000                            .WORD                       ;MWP OFF
042346  000000                            .WORD                       ;WLE OFF

042350  000070                            .WORD   RD                  ;READ DATA COMMAND
042352  000010                            .WORD   MWP                 ;MWP ON
042354  000000                            .WORD                       ;WLE OFF

042356  000020                            .WORD   RIP                 ;READ IN PRESET COMMAND
042360  000010                            .WORD   MWP                 ;MWP ON
042362  000000                            .WORD                       ;WLE OFF

042364  000000                            .WORD                       ;END OF TABLE
042366  177777                            .WORD   -1

042370                            200$:                               ;END OF TEST

                                  ;;*****************************************************************
                                  ;*TEST 111        EXCEPTION TEST

                                  ;;*****************************************************************
042370  000004                    TST111: SCOPE
042372  012737  000111  001226            MOV     #111,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX

                                  ;WITH OCCUPIED SET, EACH OF THE FOLLOWING ERRORS SHOULD CAUSE AN
                                  ;EXCEPTION:
                                  ;
                                  ;       PAR, IF RUN AND GO IS SET
                                  ;       RMR, IF RUN AND GO IS SET
                                  ;       ILR, IF RUN AND GO IS SET
                                  ;       DCK
                                  ;       HCE
                                  ;       HCRC
                                  ;       FER
                                  ;       OPI
```

```
        042400  000240                              NOP
        042402  012737  000024  001120              MOV     #20.,$ICNT      ;20 ITERATIONS
        042410  112737  000001  001131              MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
        042416  012737  042432  001122              MOV     #T111,$LPADR  ;LOAD LOOP ON TEST ADDRESS
        042424  012737  042432  001124              MOV     #T111,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
        042432                              T111:
        042432  012706  001100                      MOV     #STACK,SP       ;LOAD THE STACK POINTER
        042436  013700  001276                      MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
        042442  013701  001456                      MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
        042446  012702  042726                      MOV     #100$,R2        ;INITIALIZE TABLE POINTER
        042452                              10$:
                                            ;INITIALIZE AND VERIFY THAT EXCEPTION IS RESET
        042452  012760  000040  000010              MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
        042460  111160  000010                      MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

        042464  016037  000024  001142              MOV     RMMR1(R0),$BDDAT                ;STORE RMMR1 AT $BDDAT
        042472  042737  167777  001142              BIC     #↑CREX,$BDDAT
        042500  001410                              BEQ     20$                     ;BRANCH IF EXCEPTION IS 0
        042502  010037  001136                      MOV     R0,$BDADR
        042506  062737  000024  001136              ADD     #RMMR1,$BDADR
        042514  005037  001140                      CLR     $GDDAT
        042520  104235                              ERROR   235                     ;CANT CLEAR EXCEPTION
        042522                              20$:
                                            ;SET VOLUME VALID USING SUBROUTINE
        042522  004737  060526                      JSR     PC,SETVV                ;GO SET VOLUME VALID
        042526  000402                              BR      30$                     ;BRANCH TO 30$ IF NO ERROR
        042530  104000                              ERROR                           ;RETURN HERE IF ERROR
        042532  000474                              BR      60$
        042534                              30$:
                                            ;EXECUTE A WRITE DATA COMMAND TO SET OCCUPIED, RUN AND GO

        042534  012760  000000  000006              MOV     #0,RMDA(R0)             ;LOAD RMDA

        042542  012760  000000  000034              MOV     #0,RMDC(R0)             ;LOAD RMDC

        042550  012760  041001  000024              MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

        042556  012760  000061  000000              MOV     #WD!GO,RMCS1(R0)                ;LOAD RMCS1
        042564  012703  000002                      MOV     #2,R3
        042570                              40$:

        042570  012760  141001  000024              MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

        042576  012760  041001  000024              MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
        042604  005303                              DEC     R3
        042606  001370                              BNE     40$                     ;ISSUE 2 CLOCKS
                                            ;LOAD ERROR REGISTER WITH ENTRY FROM TABLE AND VERIFY THAT EXCEPTION IS SET

        042610  011260  000014                      MOV     (R2),RMER1(R0)  ;LOAD RMER1
        042614  012737  000200  001524              MOV     #200,WATCH      ;SET WATCHDOG TIMER VALUE
        042622  004777  136700                      JSR     PC,@CLOCK       ;START THE CLOCK
        042626                              45$:

        042626  016037  000024  001142              MOV     RMMR1(R0),$BDDAT                ;STORE RMMR1 AT $BDDAT
        042634  042737  167777  001142              BIC     #↑CREX,$BDDAT
```

```
042642 001021                          BNE     50$
042644 005737 001524                   TST     WATCH                ;HAS CLOCK EXPIRED ??
042650 001366                          BNE     45$                  ;NO - TAKE ANOTHER SAMPLE
042652 004777 136652                   JSR     PC,@STOP             ;STOP THE CLOCK
042656 012737 010000 001140            MOV     #REX,$GDDAT
042664 010037 001136                   MOV     R0,$BDADR
042670 062737 000024 001136            ADD     #RMMR1,$BDADR
042676 011237 001174                   MOV     (R2),$TMP0
042702 104236                          ERROR   236                  ;CANT SET EXCEPTION
042704 000407                          BR      60$
042706                         50$:
                               ;ADVANCE TABLE POINTER-EXIT IF DONE
042706 004777 136616                   JSR     PC,@STOP             ;STOP THE CLOCK
042712 062702 000002                   ADD     #2,R2
042716 005712                          TST     (R2)
042720 001401                          BEQ     60$
042722 000653                          BR      10$
042724 000402                  60$:    BR      200$                 ;JUMP OVER TABLE

042726                         100$:
                               ;PRESENTLY, THE TABLE HAS ONLY ONE ENTRY, THE TEST USING RMR.  THE
                               ;TABLE SHOULD BE EXPANDED TO TEST ALL THE CONDITIONS LISTED ABOVE IF
                               ;A HARDWARE CHANGE IS MADE SUCH THAT IT IS POSSIBLE TO WRITE ERROR
                               ;REGISTER 1 WITH GO SET.

042726 000004                          .WORD   RMR                  ;RMR IS CAUSED BY HARDWARE

042730 000000                          .WORD                        ;END OF TABLE
042732                         200$:                                ;END OF TEST

                               ;;*****************************************************************
                               ;*TEST 112          SET OPI TEST

                               ;;*****************************************************************
042732 000004                  TST112: SCOPE
042734 012737 000112 001226            MOV     #112,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX

042742 000240                          NOP
042744 012737 000024 001120            MOV     #20.,$ICNT           ;20 ITERATIONS
042752 112737 000001 001131            MOVB    #1,$ERMAX            ;ONE ERROR ALLOWED
042760 012737 042774 001122            MOV     #T112,$LPADR     ;LOAD LOOP ON TEST ADDRESS
042766 012737 042774 001124            MOV     #T112,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
042774                         T112:
042774 012706 001100                   MOV     #STACK,SP            ;LOAD THE STACK POINTER
043000 013700 001276                   MOV     $BASE,R0             ;R0 = UNIBUS ADDRESS OF UUT
043004 013701 001456                   MOV     TSTQUE,R1            ;R1 = POINTER TO DEVICE
043010 005002                          CLR     R2                   ;INITIALIZE FUNCTION CODE
043012 010037 001136                   MOV     R0,$BDADR            ;SETUP ERROR MSG
043016 062737 000014 001136            ADD     #RMER1,$BDADR
043024                         10$:
                               ;CLEAR AND VERIFY OPI IS RESET
043024 012760 000040 000010            MOV     #CLR,RMCS2(R0)   ;CLEAR THE MASSBUS
043032 111160 000010                   MOVB    (R1),RMCS2(R0)   ;SELECT UNIT

043036 016037 000014 001142            MOV     RMER1(R0),$BDDAT          ;STORE RMER1 AT $BDDAT
043044 042737 157777 001142            BIC     #^COPI,$BDDAT
```

```
043052  001403                          BEQ     20$                     ;BRANCH IF OPI IS ZERO
043054  005037  001140                  CLR     $GDDAT
043060  104237                          ERROR   237                     ;CANT CLEAR OPI
043062                          20$:
                                ;EXECUTE THE FUNCTION CODE AND VERIFY OPI IS SET
043062  012760  000001  000024          MOV     #DMD,RMMR1(R0)          ;LOAD RMMR1
043070  012760  040001  000024          MOV     #DMD!DBEN,RMMR1(R0)             ;LOAD RMMR1
043076  010203                          MOV     R2,R3                   ;SET GO
043100  052703  000001                  BIS     #GO,R3
043104  010360  000000                  MOV     R3,RMCS1(R0)            ;LOAD RMCS1
043110  012704  000004                  MOV     #4,R4                   ;R4=CLOCK COUNT
043114                          30$:
043114  012760  140001  000024          MOV     #DMD!DBEN!DBCK,RMMR1(R0)                ;LOAD RMMR1
043122  012760  040001  000024          MOV     #DMD!DBEN,RMMR1(R0)             ;LOAD RMMR1
043130  005304                          DEC     R4
043132  001370                          BNE     30$                     ;ISSUE 4 CLOCKS
043134  016037  000014  001142          MOV     RMER1(R0),$BDDAT                ;STORE RMER1 AT $BDDAT
043142  042737  157777  001142          BIC     #↑COPI,$BDDAT
043150  001007                          BNE     40$
043152  012737  020000  001140          MOV     #OPI,$GDDAT
043160  010237  001174                  MOV     R2,$TMP0
043164  104240                          ERROR   240                     ;CANT SET OPI
043166  000406                          BR      50$
043170                          40$:
                                ;ADVANCE FUNCTION CODE-EXIT IF DONE
043170  062702  000002                  ADD     #2,R2
043174  022702  000076                  CMP     #ILF76,R2
043200  103401                          BLO     50$
043202  000710                          BR      10$                     ;REPEAT FOR NEXT CODE
043204                          50$:                                    ;END OF TEST
                                ;;*****************************************************************
                                ;*TEST 113      RECALIBRATE TEST
                                ;;*****************************************************************
043204  000004                  TST113: SCOPE
043206  012737  000113  001226          MOV     #113,$TESTN             ;;SET TEST NUMBER IN APT MAIL BOX
043214  000240                          NOP
043216  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
043224  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
043232  012737  043246  001122          MOV     #T113,$LPADR    ;LOAD LOOP ON TEST ADDRESS
043240  012737  043246  001124          MOV     #T113,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
043246                          T113:
043246  012706  001100                  MOV     #STACK,SP               ;LOAD THE STACK POINTER
043252  013700  001276                  MOV     $BASE,R0                ;R0 = UNIBUS ADDRESS OF UUT
043256  013701  001456                  MOV     TSTQUE,R1               ;R1 = POINTER TO DEVICE
                                ;;*****************************************************************
```

```
                                        ;VERIFY THAT OPI SETS IF UNIT READY DROPS AFTER DECODE

                                        ;SET VOLUME VALID USING SUBROUTINE
043262  004737  060526                          JSR     PC,SETVV                ;GO SET VOLUME VALID
043266  000403                                  BR      10$                     ;BRANCH TO 10$ IF NO ERROR
043270  104000                                  ERROR                           ;RETURN HERE IF ERROR
043272  000137  044570                          JMP     330$
043276                                  10$:
                                        ;       ENABLE DEBUG CLOCK AND LOAD RECALIBRATE COMMAND

043276  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

043304  012760  000000  000014                  MOV     #0,RMER1(R0)            ;LOAD RMER1

043312  012760  000000  000042                  MOV     #0,RMER2(R0)            ;LOAD RMER2

043320  012760  000007  000000                  MOV     #RECAL!GO,RMCS1(R0)     ;LOAD RMCS1
                                        ;       STEP COMMAND SEQUENCER TO RECAL COM (2 CLOCKS)
043326  012702  000002                          MOV     #2,R2
043332                                  20$:

043332  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

043340  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
043346  005302                                  DEC     R2
043350  001370                                  BNE     20$
                                        ;       DROP UNIT READY AND STEP COMMAND SEQUENCER (2 CLOCKS)

043352  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1
043360  012702  000002                          MOV     #2,R2
043364                                  30$:

043364  012760  140001  000024                  MOV     #DMD!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

043372  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)     ;LOAD RMMR1
043400  005302                                  DEC     R2
043402  001370                                  BNE     30$
                                        ;       VERIFY THAT OPI IS SET

043404  016037  000014  001142                  MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
043412  042737  157777  001142                  BIC     #↑COPI,$BDDAT
043420  001011                                  BNE     40$
043422  012737  020000  001140                  MOV     #OPI,$GDDAT
043430  010037  001136                          MOV     R0,$BDADR
043434  062737  000014  001136                  ADD     #RMER1,$BDADR
043442  104241                                  ERROR   241                     ;OPI NOT SET DURING RECAL

043444  012737  043452  001124          40$:    MOV     #50$,$LPERR             ;CHANGE LOOP ON ERROR ADDRESS

                                        ;;***********************************************************
                                        ;VERIFY THAT RECALIBRATE ABORTS DURING EXECUTION

043452                                  50$:
                                        ;SET VOLUME VALID USING SUBROUTINE
043452  004737  060526                          JSR     PC,SETVV                ;GO SET VOLUME VALID
043456  000403                                  BR      60$                     ;BRANCH TO 60$ IF NO ERROR
```

```
043460  104000                          ERROR                           ;RETURN HERE IF ERROR
043462  000137  044570                  JMP     330$

043466                          60$:
                                ;       ENABLE DEBUG CLOCK AND LOAD RECALIBRATE COMMAND

043466  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

043474  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

043502  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

043510  012760  000007  000000          MOV     #RECAL!GO,RMCS1(R0)     ;LOAD RMCS1
                                ;       STEP THE COMMAND SEQUENCER TO FIRST TEST FOR ABORT (3 CLOCKS)
043516  012702  000003                  MOV     #3,R2
043522                          70$:

043522  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

043530  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
043536  005302                          DEC     R2
043540  001370                          BNE     70$
                                ;       SET DRIVE FAULT TO CAUSE ABORT CONDITION

043542  012760  041101  000024          MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
                                ;       STEP 2 CLOCKS AND VERIFY GO IS RESET
043550  012702  000002                  MOV     #2,R2
043554                          80$:

043554  012760  141101  000024          MOV     #DMD!MUR!DBEN!MDF!DBCK,RMMR1(R0)        ;LOAD RMMR1

043562  012760  041101  000024          MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
043570  005302                          DEC     R2
043572  001370                          BNE     80$

043574  016037  000000  001142          MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
043602  042737  177776  001142          BIC     #↑CGO,$BDDAT
043610  001405                          BEQ     90$
043612  010037  001136                  MOV     R0,$BDADR
043616  005037  001140                  CLR     $GDDAT
043622  104242                          ERROR   242                     ;GO NOT RESET DUE TO ABORT

043624  012737  043632  001124  90$:    MOV     #100$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS

                                ;;*****************************************************************
                                ;VERIFY OPI SETS IF ON CYLINDER LATCH DOESNT CLEAR

043632                          100$:
                                ;SET VOLUME VALID USING SUBROUTINE
043632  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
043636  000403                          BR      110$            ;BRANCH TO 110$ IF NO ERROR
043640  104000                          ERROR                   ;RETURN HERE IF ERROR
043642  000137  044570                  JMP     330$
043646                          110$:
                                ;       ENABLE DEBUG CLOCK AND LOAD RECALIBRATE COMMAND
```

```
043646 012760 041001 000024           MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
043654 012760 000000 000014           MOV     #0,RMER1(R0)      ;LOAD RMER1
043662 012760 000000 000042           MOV     #0,RMER2(R0)      ;LOAD RMER2
043670 012760 000007 000000           MOV     #RECAL!GO,RMCS1(R0)      ;LOAD RMCS1
                                ;     STEP THE COMMAND SEQUENCER
043676 012702 000017                  MOV     #15.,R2
043702                          120$:
043702 012760 141001 000024           MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1
043710 012760 041001 000024           MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
043716 005302                         DEC     R2
043720 001370                         BNE     120$
                                ;     VERIFY THAT OPI IS SET
043722 016037 000014 001142           MOV     RMER1(R0),$BDDAT         ;STORE RMER1 AT $BDDAT
043730 042737 157777 001142           BIC     #↑COPI,$BDDAT
043736 001011                         BNE     130$
043740 010037 001136                  MOV     R0,$BDADR
043744 062737 000014 001136           ADD     #RMER1,$BDADR
043752 012737 020000 001140           MOV     #OPI,$GDDAT
043760 104243                         ERROR   243               ;OPI NOT SET DUE TO ON LATCH
043762 012737 043770 001124     130$: MOV     #150$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS

                                ;;**********************************************************
                                ;VERIFY ATA SETS IF DRIVE COMPLETES RECALIBRATE (ON CYLINDER SETS)

043770                          150$:
                                ;SET VOLUME VALID USING SUBROUTINE
043770 004737 060526                  JSR     PC,SETVV          ;GO SET VOLUME VALID
043774 000403                         BR      160$              ;BRANCH TO 160$ IF NO ERROR
043776 104000                         ERROR                     ;RETURN HERE IF ERROR
044000 000137 044570                  JMP     330$
044004                          160$:
                                ;     ENABLE DEBUG CLOCK AND LOAD RECALIBRATE COMMAND
044004 012760 041401 000024           MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)      ;LOAD RMMR1
044012 012760 000000 000014           MOV     #0,RMER1(R0)      ;LOAD RMER1
044020 012760 000000 000042           MOV     #0,RMER2(R0)      ;LOAD RMER2
044026 012760 000007 000000           MOV     #RECAL!GO,RMCS1(R0)      ;LOAD RMCS1
                                ;     STEP COMMAND SEQUENCER TO FIRST ON LATCH TEST (13 CLOCKS)
044034 012702 000015                  MOV     #13.,R2
044040                          170$:
044040 012760 141401 000024           MOV     #DMD!MUR!DBEN!DBCK!MOC,RMMR1(R0)         ;LOAD RMMR1
044046 012760 041401 000024           MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)      ;LOAD RMMR1
044054 005302                         DEC     R2
044056 001370                         BNE     170$
```

```
                                   ;       DROP ON CYLINDER TO RESET LATCH, THEN SET ON CYLINDER
044060 012760 041001 000024                MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
044066 012760 041401 000024                MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
                                   ;       STEP COMMAND SEQUENCER TO SET ATTENTION (3 CLOCKS)
044074 012702 000003                       MOV     #3,R2
044100                             180$:
044100 012760 141401 000024                MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1
044106 012760 041401 000024                MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
044114 005302                              DEC     R2
044116 001370                              BNE     180$
                                   ;       VERIFY ATA IS SET
044120 016037 000012 001142                MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
044126 042737 077777 001142                BIC     #↑CATA,$BDDAT
044134 001011                              BNE     190$
044136 012737 100000 001140                MOV     #ATA,$GDDAT
044144 010037 001136                       MOV     R0,$BDADR
044150 062737 000012 001136                ADD     #RMDS,$BDADR
044156 104244                              ERROR   244             ;ATA NOT SET BY RECAL
044160 012737 044166 001124        190$:   MOV     #200$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS
                                   ;;**********************************************************
                                   ;VERIFY THAT RECALIBRATE ABORTS AFTER EXECUTION DURING WAIT LOOP
044166                             200$:
                                   ;SET VOLUME VALID USING SUBROUTINE
044166 004737 060526                       JSR     PC,SETVV        ;GO SET VOLUME VALID
044172 000402                              BR      210$            ;BRANCH TO 210$ IF NO ERROR
044174 104000                              ERROR                   ;RETURN HERE IF ERROR
044176 000574                              BR      330$
044200                             210$:
                                   ;       ENABLE DEBUG CLOCK AND LOAD RECALIBRATE COMMAND
044200 012760 041401 000024                MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
044206 012760 000000 000014                MOV     #0,RMER1(R0)    ;LOAD RMER1
044214 012760 000000 000042                MOV     #0,RMER2(R0)    ;LOAD RMER2
044222 012760 000007 000000                MOV     #RECAL!GO,RMCS1(R0)     ;LOAD RMCS1
                                   ;       STEP COMMAND SEQUENCER TO FIRST ON LATCH TEST (13 CLOCKS)
044230 012702 000015                       MOV     #13.,R2
044234                             220$:
044234 012760 141401 000024                MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1
044242 012760 041401 000024                MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
044250 005302                              DEC     R2
044252 001370                              BNE     220$
                                   ;       DROP ON CYLINDER TO RESET LATCH, LEAVE ON CYLINDER 0
```

```
044254  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                ;       STEP COMMAND SEQUENCER THROUGH WAIT LOOP (7 CLOCKS)
044262  012702  000007                  MOV     #7,R2
044266                          230$:
044266  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
044274  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
044302  005302                          DEC     R2
044304  001370                          BNE     230$
                                ;       VERIFY THAT GO IS STILL SET
044306  016037  000000  001142          MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
044314  042737  177776  001142          BIC     #↑CGO,$BDDAT
044322  001006                          BNE     240$
044324  012737  000001  001140          MOV     #GO,$GDDAT
044332  010037  001136                  MOV     R0,$BDADR
044336  104245                          ERROR   245             ;GO RESET EARLY DURING RECAL
                                ;       SET SEEK INCOMPLETE ERROR
044340                          240$:
044340  012760  041201  000024          MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)    ;LOAD RMMR1
                                ;       STEP COMMAND SEQUENCER AND VERIFY GO RESETS (3 CLOCKS)
044346  012702  000003                  MOV     #3,R2
044352                          250$:
044352  012760  141201  000024          MOV     #DMD!MUR!DBEN!MSER!DBCK,RMMR1(R0)       ;LOAD RMMR1
044360  012760  041201  000024          MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)    ;LOAD RMMR1
044366  005302                          DEC     R2
044370  001370                          BNE     250$
044372  016037  000000  001142          MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
044400  042737  177776  001142          BIC     #↑CGO,$BDDAT
044406  001405                          BEQ     260$
044410  010037  001136                  MOV     R0,$BDADR
044414  005037  001140                  CLR     $GDDAT
044420  104246                          ERROR   246             ;GO NOT RESET DUR TO WAIT ABORT
044422  012737  044430  001124  260$:   MOV     #300$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS

                                ;;*************************************************************
                                ;VERIFY THE TAG BUS DURING RECALIBRATE

044430                          300$:
                                ;SET VOLUME VALID USING SUBROUTINE
044430  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
044434  000402                          BR      310$            ;BRANCH TO 310$ IF NO ERROR
044436  104000                          ERROR                   ;RETURN HERE IF ERROR
044440  000453                          BR      330$
044442                          310$:
                                ;       ENABLE DEBUG CLOCK AND LOAD RECALIBRATE COMMAND
044442  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
044450  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
```

```

044456  012760  000000  000042          MOV     #0,RMER2(R0)        ;LOAD RMER2

044464  012760  000007  000000          MOV     #RECAL!GO,RMCS1(R0)         ;LOAD RMCS1
044472  012702  044572                  MOV     #400$,R2            ;INITIALIZE TABLE POINTER
                                ;       VERIFY TAG BUS ACCORDING TO TABLE
044476                          315$:

044476  016037  000040  001142          MOV     RMMR2(R0),$BDDAT            ;STORE RMMR2 AT $BDDAT
044504  042737  150000  001142          BIC     #RQA!RQB!TST,$BDDAT
044512  021237  001142                  CMP     (R2),$BDDAT
044516  001411                          BEQ     320$
044520  011237  001140                  MOV     (R2),$GDDAT
044524  010037  001136                  MOV     R0,$BDADR
044530  062737  000040  001136          ADD     #RMMR2,$BDADR
044536  104247                          ERROR   247                 ;INCORRECT TAG BUS DURING RECAL
044540  000413                          BR      330$
044542                          320$:
                                ;       ADVANCE TO NEXT ENTRY IN TABLE-EXIT IF DONE
044542  062702  000002                  ADD     #2,R2
044546  005712                          TST     (R2)
044550  100407                          BMI     330$                ;EXIT IF ENTRY NEGATIVE
                                ;       STEP THE COMMAND SEQUENCER AND REPEAT VERIFICATION

044552  012760  151001  000024          MOV     #DMD!DBEN!MUR!MOV!DBCK,RMMR1(R0)            ;LOAD RMMR1

044560  012760  041401  000024          MOV     #DMD!DBEN!MUR!MOC,RMMR1(R0)         ;LOAD RMMR1
044566  000743                          BR      315$
044570  000416                  330$:   BR      500$                ;JUMP OVER TABLE

044572                          400$:

                                ;TABLE OF TAG BUS CONTROL AND BIT VALUES

044572  001777                          .WORD   1777                ;BUS BITS AT HIGH IMPEDANCE STATE
044574  001777                          .WORD   1777
044576  006100                          .WORD   CC!CH!BB06          ;CONTROL BITS ENABLED, BIT 6 ON
044600  006100                          .WORD   CC!CH!BB06
044602  026100                          .WORD   TAG!CC!CH!BB06      ;TAG COMES ON
044604  026100                          .WORD   TAG!CC!CH!BB06
044606  026100                          .WORD   TAG!CC!CH!BB06
044610  026100                          .WORD   TAG!CC!CH!BB06
044612  026100                          .WORD   TAG!CC!CH!BB06
044614  026100                          .WORD   TAG!CC!CH!BB06
044616  006100                          .WORD   CC!CH!BB06          ;TAG GOES OFF
044620  006100                          .WORD   CC!CH!BB06
044622  001777                          .WORD   1777                ;CONTROL BITS DISABLED

044624  177777                          .WORD   -1                  ;END OF TABLE

044626                          500$:                               ;END OF TEST


                                ;;*****************************************************************
                                ;*TEST 114      SEEK TEST

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

```
044626  000004                            TST114: SCOPE
044630  012737  000114  001226                    MOV     #114,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

044636  000240                                    NOP
044640  012737  000024  001120                    MOV     #20.,$ICNT        ;20 ITERATIONS
044646  112737  000001  001131                    MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
044654  012737  044670  001122                    MOV     #T114,$LPADR  ;LOAD LOOP ON TEST ADDRESS
044662  012737  044670  001124                    MOV     #T114,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
044670                                    T114:
044670  012706  001100                            MOV     #STACK,SP         ;LOAD THE STACK POINTER
044674  013700  001276                            MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
044700  013701  001456                            MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE

                                          ;;*******************************************************
                                          ;VERIFY THAT OPI SETS IF UNIT READY DROPS DURING COMMAND EXECUTION

                                          ;SET VOLUME VALID USING SUBROUTINE
044704  004737  060526                            JSR     PC,SETVV          ;GO SET VOLUME VALID
044710  000403                                    BR      10$               ;BRANCH TO 10$ IF NO ERROR
044712  104000                                    ERROR                     ;RETURN HERE IF ERROR
044714  000137  046334                            JMP     330$
044720                                    10$:
                                          ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                          ;       ADDRESS,  AND LOAD SEEK COMMAND

044720  012760  041001  000024                    MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

044726  012760  000000  000006                    MOV     #0,RMDA(R0)       ;LOAD RMDA

044734  012760  000000  000034                    MOV     #0,RMDC(R0)       ;LOAD RMDC

044742  012760  000000  000014                    MOV     #0,RMER1(R0)      ;LOAD RMER1

044750  012760  000000  000042                    MOV     #0,RMER2(R0)      ;LOAD RMER2

044756  012760  000005  000000                    MOV     #SEEK!GO,RMCS1(R0)        ;LOAD RMCS1
                                          ;       STEP COMMAND SEQUENCER TO SEEK COM (2 CLOCKS)
044764  012702  000002                            MOV     #2,R2
044770                                    20$:

044770  012760  141001  000024                    MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1

044776  012760  041001  000024                    MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
045004  005302                                    DEC     R2
045006  001370                                    BNE     20$
                                          ;       DROP UNIT READY AND STEP COMMAND SEQUENCER (2 CLOCKS)

045010  012760  040001  000024                    MOV     #DMD!DBEN,RMMR1(R0)       ;LOAD RMMR1
045016  012702  000002                            MOV     #2,R2
045022                                    30$:

045022  012760  140001  000024                    MOV     #DMD!DBEN!DBCK,RMMR1(R0)         ;LOAD RMMR1

045030  012760  040001  000024                    MOV     #DMD!DBEN,RMMR1(R0)       ;LOAD RMMR1
045036  005302                                    DEC     R2
045040  001370                                    BNE     30$
```

```
                                        ;       VERIFY THAT OPI IS SET
045042  016037  000014  001142                  MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
045050  042737  157777  001142                  BIC     #↑COPI,$BDDAT
045056  001011                                  BNE     40$
045060  012737  020000  001140                  MOV     #OPI,$GDDAT
045066  010037  001136                          MOV     R0,$BDADR
045072  062737  000014  001136                  ADD     #RMER1,$BDADR
045100  104250                                  ERROR   250             ;OPI NOT SET DURING SEEK
045102  012737  045110  001124          40$:    MOV     #50$,$LPERR     ;CHANGE LOOP ON ERROR ADDRESS
                                        ;;****************************************************
                                        ;VERIFY THAT SEEK ABORTS DURING EXECUTION
045110                                  50$:
                                        ;SET VOLUME VALID USING SUBROUTINE
045110  004737  060526                          JSR     PC,SETVV        ;GO SET VOLUME VALID
045114  000403                                  BR      60$             ;BRANCH TO 60$ IF NO ERROR
045116  104000                                  ERROR                   ;RETURN HERE IF ERROR
045120  000137  046334                          JMP     330$
045124                                  60$:
                                        ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                        ;       ADDRESS,  AND LOAD SEEK COMMAND
045124  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
045132  012760  000000  000006                  MOV     #0,RMDA(R0)     ;LOAD RMDA
045140  012760  000000  000034                  MOV     #0,RMDC(R0)     ;LOAD RMDC
045146  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1
045154  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2
045162  012760  000005  000000                  MOV     #SEEK!GO,RMCS1(R0)      ;LOAD RMCS1
                                        ;       STEP THE COMMAND SEQUENCER TO FIRST TEST FOR ABORT (3 CLOCKS)
045170  012702  000003                          MOV     #3,R2
045174                                  70$:
045174  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
045202  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
045210  005302                                  DEC     R2
045212  001370                                  BNE     70$
                                        ;       SET DRIVE FAULT TO CAUSE ABORT CONDITION
045214  012760  041101  000024                  MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
                                        ;       STEP 2 CLOCKS AND VERIFY GO IS RESET
045222  012702  000002                          MOV     #2,R2
045226                                  80$:
045226  012760  141101  000024                  MOV     #DMD!MUR!DBEN!MDF!DBCK,RMMR1(R0)        ;LOAD RMMR1
045234  012760  041101  000024                  MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
```

```
045242  005302                          DEC     R2
045244  001370                          BNE     80$

045246  016037  000000  001142          MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
045254  042737  177776  001142          BIC     #↑CGO,$BDDAT
045262  001405                          BEQ     90$
045264  010037  001136                  MOV     R0,$BDADR
045270  005037  001140                  CLR     $GDDAT
045274  104251                          ERROR   251             ;GO NOT RESET DUE TO ABORT

045276  012737  045304  001124  90$:    MOV     #100$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS

                                ;;************************************************************
                                ;VERIFY OPI SETS IF ON CYLINDER LATCH DOESNT CLEAR

045304                          100$:
                                ;SET VOLUME VALID USING SUBROUTINE
045304  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
045310  000403                          BR      110$            ;BRANCH TO 110$ IF NO ERROR
045312  104000                          ERROR                   ;RETURN HERE IF ERROR
045314  000137  046334                  JMP     330$
045320                          110$:
                                ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                ;       ADDRESS,  AND LOAD SEEK COMMAND

045320  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

045326  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA

045334  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC

045342  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

045350  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

045356  012760  000005  000000          MOV     #SEEK!GO,RMCS1(R0)      ;LOAD RMCS1
                                ;       STEP THE COMMAND SEQUENCER
045364  012702  000017                  MOV     #15.,R2
045370                          120$:

045370  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

045376  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
045404  005302                          DEC     R2
045406  001370                          BNE     120$
                                ;       VERIFY THAT OPI IS SET

045410  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
045416  042737  157777  001142          BIC     #↑COPI,$BDDAT
045424  001011                          BNE     130$
045426  010037  001136                  MOV     R0,$BDADR
045432  062737  000014  001136          ADD     #RMER1,$BDADR
045440  012737  020000  001140          MOV     #OPI,$GDDAT
045446  104252                          ERROR   252             ;OPI NOT SET DUE TO ON LATCH

045450  012737  045456  001124  130$:   MOV     #150$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS
```

```
                                        ;;***************************************************************
                                        ;VERIFY ATA SETS IF DRIVE COMPLETES SEEK (ON CYLINDER SETS)

045456                                  150$:
                                        ;SET VOLUME VALID USING SUBROUTINE
045456  004737  060526                          JSR     PC,SETVV                ;GO SET VOLUME VALID
045462  000403                                  BR      160$                    ;BRANCH TO 160$ IF NO ERROR
045464  104000                                  ERROR                           ;RETURN HERE IF ERROR
045466  000137  046334                          JMP     330$
045472                                  160$:
                                        ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                        ;       ADDRESS,  AND LOAD SEEK COMMAND

045472  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1

045500  012760  000000  000006                  MOV     #0,RMDA(R0)     ;LOAD RMDA

045506  012760  000000  000034                  MOV     #0,RMDC(R0)     ;LOAD RMDC

045514  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1

045522  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

045530  012760  000005  000000                  MOV     #SEEK!GO,RMCS1(R0)      ;LOAD RMCS1
                                        ;       STEP COMMAND SEQUENCER TO FIRST ON LATCH TEST (13 CLOCKS)
045536  012702  000015                          MOV     #13.,R2
045542                                  170$:

045542  012760  141401  000024                  MOV     #DMD!MUR!DBEN!DBCK!MOC,RMMR1(R0)        ;LOAD RMMR1

045550  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
045556  005302                                  DEC     R2
045560  001370                                  BNE     170$
                                        ;       DROP ON CYLINDER TO RESET LATCH, THEN SET ON CYLINDER

045562  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

045570  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
                                        ;       STEP COMMAND SEQUENCER TO SET ATTENTION (3 CLOCKS)
045576  012702  000003                          MOV     #3,R2
045602                                  180$:

045602  012760  141401  000024                  MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1

045610  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
045616  005302                                  DEC     R2
045620  001370                                  BNE     180$
                                        ;       VERIFY ATA IS SET

045622  016037  000012  001142                  MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
045630  042737  077777  001142                  BIC     #↑CATA,$BDDAT
045636  001011                                  BNE     190$
045640  012737  100000  001140                  MOV     #ATA,$GDDAT
045646  010037  001136                          MOV     R0,$BDADR
045652  062737  000012  001136                  ADD     #RMDS,$BDADR
```

```
045660  104253                                  ERROR   253                     ;ATA NOT SET BY SEEK
045662  012737  045670  001124          190$:   MOV     #200$,$LPERR            ;CHANGE LOOP ON ERROR ADDRESS
                                        ;;**************************************************************
                                        ;VERIFY THAT SEEK ABORTS AFTER EXECUTION DURING WAIT LOOP
045670                                  200$:
                                        ;SET VOLUME VALID USING SUBROUTINE
045670  004737  060526                          JSR     PC,SETVV                ;GO SET VOLUME VALID
045674  000403                                  BR      210$                    ;BRANCH TO 210$ IF NO ERROR
045676  104000                                  ERROR                           ;RETURN HERE IF ERROR
045700  000137  046334                          JMP     330$
045704                                  210$:
                                        ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                        ;       ADDRESS  AND LOAD SEEK COMMAND
045704  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
045712  012760  000000  000006                  MOV     #0,RMDA(R0)             ;LOAD RMDA
045720  012760  000000  000034                  MOV     #0,RMDC(R0)             ;LOAD RMDC
045726  012760  000000  000014                  MOV     #0,RMER1(R0)            ;LOAD RMER1
045734  012760  000000  000042                  MOV     #0,RMER2(R0)            ;LOAD RMER2
045742  012760  000005  000000                  MOV     #SEEK!GO,RMCS1(R0)      ;LOAD RMCS1
                                        ;       STEP COMMAND SEQUENCER TO FIRST ON LATCH TEST (13 CLOCKS)
045750  012702  000015                          MOV     #13.,R2
045754                                  220$:
045754  012760  141401  000024                  MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1
045762  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
045770  005302                                  DEC     R2
045772  001370                                  BNE     220$
                                        ;       DROP ON CYLINDER TO RESET LATCH, LEAVE ON CYLINDER 0
045774  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                        ;       STEP COMMAND SEQUENCER THROUGH WAIT LOOP (7 CLOCKS)
046002  012702  000007                          MOV     #7,R2
046006                                  230$:
046006  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
046014  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
046022  005302                                  DEC     R2
046024  001370                                  BNE     230$
                                        ;       VERIFY THAT GO IS STILL SET
046026  016037  000000  001142                  MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
046034  042737  177776  001142                  BIC     #↑CGO,$BDDAT
046042  001006                                  BNE     240$
046044  012737  000001  001140                  MOV     #GO,$GDDAT
046052  010037  001136                          MOV     R0,$BDADR
```

```
046056  104254                         ERROR   254                     ;GO RESET EARLY DURING SEEK
                               ;       SET SEEK INCOMPLETE ERROR
046060                         240$:

046060  012760  041201  000024         MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)     ;LOAD RMMR1
                               ;       STEP COMMAND SEQUENCER AND VERIFY GO RESETS (3 CLOCKS)
046066  012702  000003                 MOV     #3,R2
046072                         250$:

046072  012760  141201  000024         MOV     #DMD!MUR!DBEN!MSER!DBCK,RMMR1(R0)        ;LOAD RMMR1

046100  012760  041201  000024         MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)     ;LOAD RMMR1
046106  005302                         DEC     R2
046110  001370                         BNE     250$

046112  016037  000000  001142         MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
046120  042737  177776  001142         BIC     #↑CGO,$BDDAT
046126  001405                         BEQ     260$
046130  010037  001136                 MOV     R0,$BDADR
046134  005037  001140                 CLR     $GDDAT
046140  104255                         ERROR   255                     ;GO NOT RESET DUE TO WAIT ABORT

046142  012737  046154  001124 260$:   MOV     #300$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS
046150  012703  000001                 MOV     #1,R3           ;INITIALIZE CYLINDER ADDRESS

                               ;;************************************************************
                               ;VERIFY THE TAG BUS DURING SEEK

046154                         300$:
                               ;SET VOLUME VALID USING SUBROUTINE
046154  004737  060526                 JSR     PC,SETVV        ;GO SET VOLUME VALID
046160  000402                         BR      310$            ;BRANCH TO 310$ IF NO ERROR
046162  104000                         ERROR                   ;RETURN HERE IF ERROR
046164  000463                         BR      330$
046166                         310$:
                               ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                               ;       ADDRESS  AND LOAD SEEK COMMAND

046166  012760  041401  000024         MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)      ;LOAD RMMR1

046174  012760  000000  000006         MOV     #0,RMDA(R0)     ;LOAD RMDA

046202  010360  000034                 MOV     R3,RMDC(R0)     ;LOAD RMDC

046206  012760  000000  000014         MOV     #0,RMER1(R0)    ;LOAD RMER1

046214  012760  000000  000042         MOV     #0,RMER2(R0)    ;LOAD RMER2

046222  012760  000005  000000         MOV     #SEEK!GO,RMCS1(R0)      ;LOAD RMCS1
046230  012702  046350                 MOV     #400$,R2        ;INITIALIZE TABLE POINTER
                               ;       VERIFY TAG BUS ACCORDING TO TABLE AND CYLINDER IN R3
046234                         315$:

046234  016037  000040  001142         MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
046242  042737  150000  001142         BIC     #RQA!RQB!TST,$BDDAT
046250  011237  001140                 MOV     (R2),$GDDAT
```

```
046254  050337  001140                          BIS     R3,$GDDAT               ;OR CYLINDER ADDRESS IN
046260  023737  001140  001142                  CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED AND RECEIVED
046266  001407                                  BEQ     320$                    ;BRANCH IF TAG BUS OK
046270  010037  001136                          MOV     R0,$BDADR
046274  062737  000040  001136                  ADD     #RMMR2,$BDADR
046302  104256                                  ERROR   256                     ;INCORRECT TAG BUS DURING SEEK
046304  000413                                  BR      330$
046306                                  320$:
                                        ;       ADVANCE TO NEXT ENTRY IN TABLE-EXIT IF DONE
046306  062702  000002                          ADD     #2,R2
046312  005712                                  TST     (R2)
046314  100407                                  BMI     330$                    ;EXIT IF ENTRY NEGATIVE
                                        ;       STEP THE COMMAND SEQUENCER AND REPEAT VERIFICATION
046316  012760  151001  000024                  MOV     #DMD!DBEN!MUR!MOV!DBCK,RMMR1(R0)        ;LOAD RMMR1
046324  012760  041401  000024                  MOV     #DMD!DBEN!MUR!MOC,RMMR1(R0)     ;LOAD RMMR1
046332  000740                                  BR      315$
046334                                  330$:
                                        ;REPEAT TAG BUS TEST FOR EACH PRIME CYLINDER, I.E., 1,2,4,...
046334  006303                                  ASL     R3                      ;SHIFT TO NEXT CYLINDER
046336  020327  000512                          CMP     R3,#512
046342  101001                                  BHI     340$                    ;EXIT IF 512 WAS DONE
046344  000703                                  BR      300$                    ;TEST NEXT CYLINDER
046346  000416                          340$:   BR      500$                    ;JUMP OVER TABLE
046350                                  400$:
                                        ;TABLE OF TAG BUS CONTROL AND BIT VALUES
046350  001777                                  .WORD   1777                    ;BUS BITS AT HIGH IMPEDANCE STATE
046352  001777                                  .WORD   1777
046354  004000                                  .WORD   CC      ;CONTROL BITS ENABLED, BIT 6 ON
046356  004000                                  .WORD   CC
046360  024000                                  .WORD   TAG!CC  ;TAG COMES ON
046362  024000                                  .WORD   TAG!CC
046364  024000                                  .WORD   TAG!CC
046366  024000                                  .WORD   TAG!CC
046370  024000                                  .WORD   TAG!CC
046372  024000                                  .WORD   TAG!CC
046374  004000                                  .WORD   CC      ;TAG GOES OFF
046376  004000                                  .WORD   CC
046400  001777                                  .WORD   1777                    ;CONTROL BITS DISABLED
046402  177777                                  .WORD   -1                      ;END OF TABLE
046404                                  500$:                                   ;END OF TEST
                                        ;;*****************************************************
                                        ;*TEST 115              SEARCH TEST
                                        ;;*****************************************************
046404  000004                          TST115: SCOPE
046406  012737  000115  001226                  MOV     #115,$TESTN             ;;SET TEST NUMBER IN APT MAIL BOX
```

```
046414  000240                                  NOP
046416  012737  000024  001120                  MOV     #20.,$ICNT          ;20 ITERATIONS
046424  112737  000001  001131                  MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
046432  012737  046446  001122                  MOV     #T115,$LPADR   ;LOAD LOOP ON TEST ADDRESS
046440  012737  046446  001124                  MOV     #T115,$LPERR   ;LOAD LOOP ON ERROR ADDRESS
046446                                  T115:
046446  012706  001100                          MOV     #STACK,SP           ;LOAD THE STACK POINTER
046452  013700  001276                          MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
046456  013701  001456                          MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
                                        ;;*************************************************************
                                        ;VERIFY THAT OPI SETS IF UNIT READY DROPS DURING COMMAND EXECUTION
                                        ;SET VOLUME VALID USING SUBROUTINE
046462  004737  060526                          JSR     PC,SETVV            ;GO SET VOLUME VALID
046466  000403                                  BR      10$                 ;BRANCH TO 10$ IF NO ERROR
046470  104000                                  ERROR                       ;RETURN HERE IF ERROR
046472  000137  050504                          JMP     330$
046476                                  10$:
                                        ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                        ;       ADDRESS,  AND LOAD SEARCH COMMAND
046476  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
046504  012760  000000  000006                  MOV     #0,RMDA(R0)         ;LOAD RMDA
046512  012760  000000  000034                  MOV     #0,RMDC(R0)         ;LOAD RMDC
046520  012760  000000  000014                  MOV     #0,RMER1(R0)        ;LOAD RMER1
046526  012760  000000  000042                  MOV     #0,RMER2(R0)        ;LOAD RMER2
046534  012760  000031  000000                  MOV     #SEARCH!GO,RMCS1(R0)     ;LOAD RMCS1
                                        ;       STEP COMMAND SEQUENCER TO SEARCH COM (2 CLOCKS)
046542  012702  000002                          MOV     #2,R2
046546                                  20$:
046546  012760  141001  000024                  MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)     ;LOAD RMMR1
046554  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
046562  005302                                  DEC     R2
046564  001370                                  BNE     20$
                                        ;       DROP UNIT READY AND STEP COMMAND SEQUENCER (2 CLOCKS)
046566  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)      ;LOAD RMMR1
046574  012702  000002                          MOV     #2,R2
046600                                  30$:
046600  012760  140001  000024                  MOV     #DMD!DBEN!DBCK,RMMR1(R0)         ;LOAD RMMR1
046606  012760  040001  000024                  MOV     #DMD!DBEN,RMMR1(R0)      ;LOAD RMMR1
046614  005302                                  DEC     R2
046616  001370                                  BNE     30$
                                        ;       VERIFY THAT OPI IS SET
046620  016037  000014  001142                  MOV     RMER1(R0),$BDDAT         ;STORE RMER1 AT $BDDAT
```

```
046626  042737  157777  001142          BIC     #↑COPI,$BDDAT
046634  001011                          BNE     40$
046636  012737  020000  001140          MOV     #OPI,$GDDAT
046644  010037  001136                  MOV     R0,$BDADR
046650  062737  000014  001136          ADD     #RMER1,$BDADR
046656  104257                          ERROR   257                     ;OPI NOT SET DURING SEARCH
046660  012737  046666  001124  40$:    MOV     #50$,$LPERR             ;CHANGE LOOP ON ERROR ADDRESS
                                ;;*****************************************************************
                                ;VERIFY THAT SEARCH ABORTS DURING EXECUTION
046666                          50$:
                                ;SET VOLUME VALID USING SUBROUTINE
046666  004737  060526                  JSR     PC,SETVV                ;GO SET VOLUME VALID
046672  000403                          BR      60$                     ;BRANCH TO 60$ IF NO ERROR
046674  104000                          ERROR                           ;RETURN HERE IF ERROR
046676  000137  050504                  JMP     330$
046702                          60$:
                                ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                ;       ADDRESS,  AND LOAD SEARCH COMMAND
046702  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
046710  012760  000000  000006          MOV     #0,RMDA(R0)             ;LOAD RMDA
046716  012760  000000  000034          MOV     #0,RMDC(R0)             ;LOAD RMDC
046724  012760  000000  000014          MOV     #0,RMER1(R0)            ;LOAD RMER1
046732  012760  000000  000042          MOV     #0,RMER2(R0)            ;LOAD RMER2
046740  012760  000031  000000          MOV     #SEARCH!GO,RMCS1(R0)    ;LOAD RMCS1
                                ;       STEP THE COMMAND SEQUENCER TO FIRST TEST FOR ABORT (3 CLOCKS)
046746  012702  000003                  MOV     #3,R2
046752                          70$:
046752  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
046760  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
046766  005302                          DEC     R2
046770  001370                          BNE     70$
                                ;       SET DRIVE FAULT TO CAUSE ABORT CONDITION
046772  012760  041101  000024          MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
                                ;       STEP 2 CLOCKS AND VERIFY GO IS RESET
047000  012702  000002                  MOV     #2,R2
047004                          80$:
047004  012760  141101  000024          MOV     #DMD!MUR!DBEN!MDF!DBCK,RMMR1(R0)        ;LOAD RMMR1
047012  012760  041101  000024          MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
047020  005302                          DEC     R2
047022  001370                          BNE     80$
```

```
047024  016037  000000  001142          MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
047032  042737  177776  001142          BIC     #↑CGO,$BDDAT
047040  001405                          BEQ     90$
047042  010037  001136                  MOV     R0,$BDADR
047046  005037  001140                  CLR     $GDDAT
047052  104260                          ERROR   260             ;GO NOT RESET DUE TO ABORT

                                ;(THE OTHER TWO ABORT TESTS IN THE COMMAND SEQUENCER ARE TESTED
                                ;DURING DATA COMMAND TESTS)

047054  012737  047062  001124  90$:    MOV     #100$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS

                                ;;*************************************************************
                                ;VERIFY OPI SETS IF ON CYLINDER LATCH DOESNT CLEAR

047062                          100$:
                                ;SET VOLUME VALID USING SUBROUTINE
047062  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
047066  000403                          BR      110$            ;BRANCH TO 110$ IF NO ERROR
047070  104000                          ERROR                   ;RETURN HERE IF ERROR
047072  000137  050504                  JMP     330$
047076                          110$:
                                ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                ;       ADDRESS,  AND LOAD SEARCH COMMAND

047076  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

047104  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA

047112  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC

047120  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

047126  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

047134  012760  000031  000000          MOV     #SEARCH!GO,RMCS1(R0)    ;LOAD RMCS1
                                ;       STEP THE COMMAND SEQUENCER
047142  012702  000023                  MOV     #19.,R2
047146                          120$:

047146  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

047154  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
047162  005302                          DEC     R2
047164  001370                          BNE     120$
                                ;       VERIFY THAT OPI IS SET

047166  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
047174  042737  157777  001142          BIC     #↑COPI,$BDDAT
047202  001011                          BNE     130$
047204  010037  001136                  MOV     R0,$BDADR
047210  062737  000014  001136          ADD     #RMER1,$BDADR
047216  012737  020000  001140          MOV     #OPI,$GDDAT
047224  104261                          ERROR   261             ;OPI NOT SET DUE TO ON LATCH

047226  012737  047234  001124  130$:   MOV     #150$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS
```

```
                                        ;;*************************************************************
                                        ;VERIFY ATA SETS IF DRIVE COMPLETES SEARCH (ON CYLINDER AND
                                        ;SECTOR COMPARE SETS)

047234                                  150$:
                                        ;SET VOLUME VALID USING SUBROUTINE
047234  004737  060526                          JSR     PC,SETVV        ;GO SET VOLUME VALID
047240  000403                                  BR      160$            ;BRANCH TO 160$ IF NO ERROR
047242  104000                                  ERROR                   ;RETURN HERE IF ERROR
047244  000137  050504                          JMP     330$
047250                                  160$:
                                        ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                        ;       ADDRESS,  AND LOAD SEARCH COMMAND

047250  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1

047256  012760  000000  000006                  MOV     #0,RMDA(R0)     ;LOAD RMDA

047264  012760  000000  000034                  MOV     #0,RMDC(R0)     ;LOAD RMDC

047272  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1

047300  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2

047306  012760  000031  000000                  MOV     #SEARCH!GO,RMCS1(R0)    ;LOAD RMCS1
                                        ;       STEP COMMAND SEQUENCER TO FIRST ON LATCH TEST (17 CLOCKS)
047314  012702  000021                          MOV     #17.,R2
047320                                  170$:

047320  012760  141401  000024                  MOV     #DMD!MUR!DBEN!DBCK!MOC,RMMR1(R0)        ;LOAD RMMR1

047326  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
047334  005302                                  DEC     R2
047336  001370                                  BNE     170$
                                        ;       DROP ON CYLINDER TO RESET LATCH,AND RAISE INDEX PULSE
                                        ;TO SET FORMAT CHANGE FLOP


047340  012760  041005  000024                  MOV     #DMD!MUR!DBEN!MI,RMMR1(R0)      ;LOAD RMMR1
                                        ;       RAISE ON CYLINDER AND INHIBIT SEARCH TIMEOUT

047346  012760  051401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
                                        ;       STEP COMMAND SEQUENCER TO SEARCH ENABLE (2 CLOCKS)
047354  012702  000002                          MOV     #2,R2
047360                                  180$:

047360  012760  151401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO!DBCK,RMMR1(R0)    ;LOAD RMMR1

047366  012760  051401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
047374  005302                                  DEC     R2
047376  001370                                  BNE     180$
                                        ;FORCE SECTOR COMPARE BY CLOCKING SECTOR PULSE WITH SECTOR COMPARE
                                        ;ACTIVE

047400  012760  051403  000024                  MOV     #DMD!MUR!MOC!DBEN!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
```

```

047406  012760  051443  000024          MOV     #DMD!MUR!MOC!DBEN!DTO!MSC!MS,RMMR1(R0)   ;LOAD RMMR1

047414  012760  051403  000024          MOV     #DMD!MUR!MOC!DBEN!DTO!MSC,RMMR1(R0)      ;LOAD RMMR1
                                ;CLOCK SEQUENCER TO SET ATA (3 CLOCKS)
047422  012702  000003                  MOV     #3,R2                   ;R2 = CLOCK COUNT
047426                          185$:

047426  012760  151401  000024          MOV     #DMD!MUR!MOC!DBEN!DTO!DBCK,RMMR1(R0)     ;LOAD RMMR1

047434  012760  051401  000024          MOV     #DMD!MUR!MOC!DBEN!DTO,RMMR1(R0) ;LOAD RMMR1
047442  005302                          DEC     R2
047444  001370                          BNE     185$
                                ;       VERIFY ATA IS SET

047446  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
047454  042737  077777  001142          BIC     #↑CATA,$BDDAT
047462  001011                          BNE     190$
047464  012737  100000  001140          MOV     #ATA,$GDDAT
047472  010037  001136                  MOV     R0,$BDADR
047476  062737  000012  001136          ADD     #RMDS,$BDADR
047504  104262                          ERROR   262                     ;ATA NOT SET BY SEARCH

047506  012737  047514  001124  190$:   MOV     #200$,$LPERR            ;CHANGE LOOP ON ERROR ADDRESS

                                ;;*************************************************************
                                ;VERIFY THAT SEARCH ABORTS AFTER EXECUTION DURING SEARCH SEEK LOOP

047514                          200$:
                                ;SET VOLUME VALID USING SUBROUTINE
047514  004737  060526                  JSR     PC,SETVV                ;GO SET VOLUME VALID
047520  000403                          BR      210$                    ;BRANCH TO 210$ IF NO ERROR
047522  104000                          ERROR                           ;RETURN HERE IF ERROR
047524  000137  050504                  JMP     330$
047530                          210$:
                                ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                ;       ADDRESS  AND LOAD SEARCH COMMAND

047530  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)      ;LOAD RMMR1

047536  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA

047544  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC

047552  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

047560  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

047566  012760  000031  000000          MOV     #SEARCH!GO,RMCS1(R0)    ;LOAD RMCS1
                                ;       STEP COMMAND SEQUENCER TO FIRST ON LATCH TEST (17 CLOCKS)
047574  012702  000021                  MOV     #17.,R2
047600                          220$:

047600  012760  141401  000024          MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)         ;LOAD RMMR1

047606  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)      ;LOAD RMMR1
```

```
 047614  005302                             DEC     R2
 047616  001370                             BNE     220$
                                    ;       DROP ON CYLINDER TO RESET LATCH, LEAVE ON CYLINDER 0

 047620  012760  041001  000024             MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                    ;       STEP COMMAND SEQUENCER THROUGH WAIT LOOP (7 CLOCKS)
 047626  012702  000007                     MOV     #7,R2
 047632                             230$:

 047632  012760  141001  000024             MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1

 047640  012760  041001  000024             MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
 047646  005302                             DEC     R2
 047650  001370                             BNE     230$
                                    ;       VERIFY THAT GO IS STILL SET

 047652  016037  000000  001142             MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
 047660  042737  177776  001142             BIC     #↑CGO,$BDDAT
 047666  001006                             BNE     240$
 047670  012737  000001  001140             MOV     #GO,$GDDAT
 047676  010037  001136                     MOV     R0,$BDADR
 047702  104263                             ERROR   263             ;GO RESET EARLY DURING SEARCH
                                    ;       SET SEEK INCOMPLETE ERROR
 047704                             240$:

 047704  012760  041201  000024             MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)    ;LOAD RMMR1
                                    ;       STEP COMMAND SEQUENCER AND VERIFY GO RESETS (3 CLOCKS)
 047712  012702  000003                     MOV     #3,R2
 047716                             250$:

 047716  012760  141201  000024             MOV     #DMD!MUR!DBEN!MSER!DBCK,RMMR1(R0)       ;LOAD RMMR1

 047724  012760  041201  000024             MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)    ;LOAD RMMR1
 047732  005302                             DEC     R2
 047734  001370                             BNE     250$

 047736  016037  000000  001142             MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
 047744  042737  177776  001142             BIC     #↑CGO,$BDDAT
 047752  001405                             BEQ     260$
 047754  010037  001136                     MOV     R0,$BDADR
 047760  005037  001140                     CLR     $GDDAT
 047764  104264                             ERROR   264             ;GO NOT RESET DUE TO WAIT ABORT

 047766  012737  047774  001124     260$:   MOV     #265$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS

                                    ;;*************************************************************
                                    ;VERIFY THAT SEARCH ABORTS DURING SECTOR COMPARE LOOP

 047774                             265$:
                                    ;SET VOLUME VALID USING SUBROUTINE
 047774  004737  060526                     JSR     PC,SETVV        ;GO SET VOLUME VALID
 050000  000403                             BR      270$            ;BRANCH TO 270$ IF NO ERROR
 050002  104000                             ERROR                   ;RETURN HERE IF ERROR
 050004  000137  050504                     JMP     330$
 050010                             270$:
                                    ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
```

```
                                        ;       ADDRESS,  AND LOAD SEARCH COMMAND
050010  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
050016  012760  000000  000006                  MOV     #0,RMDA(R0)     ;LOAD RMDA
050024  012760  000000  000034                  MOV     #0,RMDC(R0)     ;LOAD RMDC
050032  012760  000000  000014                  MOV     #0,RMER1(R0)    ;LOAD RMER1
050040  012760  000000  000042                  MOV     #0,RMER2(R0)    ;LOAD RMER2
050046  012760  000031  000000                  MOV     #SEARCH!GO,RMCS1(R0)    ;LOAD RMCS1
                                        ;       STEP COMMAND SEQUENCER TO FIRST ON LATCH TEST (17 CLOCKS)
050054  012702  000021                          MOV     #17.,R2
050060                                  275$:
050060  012760  141401  000024                  MOV     #DMD!MUR!DBEN!DBCK!MOC,RMMR1(R0)        ;LOAD RMMR1
050066  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
050074  005302                                  DEC     R2
050076  001370                                  BNE     275$
                                        ;       DROP ON CYLINDER TO RESET LATCH, AND RAISE INDEX PULSE
                                        ;TO SET FORMAT CHANGE FLOP
050100  012760  041005  000024                  MOV     #DMD!MUR!DBEN!MI,RMMR1(R0)      ;LOAD RMMR1
                                        ;       RAISE ON CYLINDER AND INHIBIT SEARCH TIMEOUT
050106  012760  051401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
                                        ;       STEP COMMAND SEQUENCER TO SEARCH ENABLE (2 CLOCKS)
050114  012702  000002                          MOV     #2,R2
050120                                  280$:
050120  012760  151401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO!DBCK,RMMR1(R0)    ;LOAD RMMR1
050126  012760  051401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
050134  005302                                  DEC     R2
050136  001370                                  BNE     280$
                                        ;       VERIFY THAT SEARCH ENABLE IS ON DURING SECTOR COMPARE LOOP
050140  012702  000004                          MOV     #4,R2           ;R2 = CLOCK COUNT
050144                                  281$:
050144  016037  000024  001142                  MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
050152  042737  173777  001142                  BIC     #^CESRC,$BDDAT
050160  001411                                  BEQ     282$            ;BRANCH IF SEARCH NOT ENABLED
050162  012760  151401  000024                  MOV     #DMD!MUR!MOC!DBEN!DTO!DBCK,RMMR1(R0)    ;LOAD RMMR1
050170  012760  051401  000024                  MOV     #DMD!MUR!MOC!DBEN!DTO,RMMR1(R0) ;LOAD RMMR1
050176  005302                                  DEC     R2
050200  001361                                  BNE     281$
050202  000411                                  BR      283$
050204  012737  004000  001140          282$:   MOV     #ESRC,$GDDAT
050212  010037  001136                          MOV     R0,$BDADR
050216  062737  000024  001136                  ADD     #RMMR1,$BDADR
```

```
050224  104265                                  ERROR   265                     ;SEARCH NOT ENABLED
050226                                  283$:
                                        ;       SET DRIVE FAULT TO CAUSE ABORT CONDITION

050226  012760  051501  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO!MDF,RMMR1(R0)     ;LOAD RMMR1
                                        ;       STEP 2 CLOCKS AND VERIFY GO IS RESET
050234  012702  000002                          MOV     #2,R2                   ;R2 = CLOCK COUNT
050240                                  285$:

050240  012760  151501  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO!MDF!DBCK,RMMR1(R0)        ;LOAD RMMR1

050246  012760  051501  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO!MDF,RMMR1(R0)     ;LOAD RMMR1
050254  005302                                  DEC     R2
050256  001370                                  BNE     285$

050260  016037  000000  001142                  MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
050266  042737  177776  001142                  BIC     #^CGO,$BDDAT
050274  001406                                  BEQ     290$
050276  012737  000000  001140                  MOV     #0,$GDDAT
050304  010037  001136                          MOV     R0,$BDADR
050310  104260                                  ERROR   260                     ;GO NOT RESET DURING SECTOR COMPARE

050312  012737  050324  001124  290$:           MOV     #300$,$LPERR            ;CHANGE LOOP ON ERROR ADDRESS
050320  012703  000001                          MOV     #1,R3                   ;INITIALIZE CYLINDER ADDRESS

                                        ;;*************************************************************
                                        ;VERIFY THE TAG BUS DURING SEARCH

050324                                  300$:
                                        ;SET VOLUME VALID USING SUBROUTINE
050324  004737  060526                          JSR     PC,SETVV                ;GO SET VOLUME VALID
050330  000402                                  BR      310$                    ;BRANCH TO 310$ IF NO ERROR
050332  104000                                  ERROR                           ;RETURN HERE IF ERROR
050334  000463                                  BR      330$
050336                                  310$:
                                        ;       ENABLE DEBUG CLOCK, LOAD CYLINDER, TRACK AND SECTOR
                                        ;       ADDRESS  AND LOAD SEARCH COMMAND

050336  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1

050344  012760  000000  000006                  MOV     #0,RMDA(R0)             ;LOAD RMDA

050352  010360  000034                          MOV     R3,RMDC(R0)             ;LOAD RMDC
```

```
 050356  012760  000000  000014          MOV     #0,RMER1(R0)       ;LOAD RMER1

 050364  012760  000000  000042          MOV     #0,RMER2(R0)       ;LOAD RMER2

 050372  012760  000031  000000          MOV     #SEARCH!GO,RMCS1(R0)     ;LOAD RMCS1
 050400  012702  050520                  MOV     #400$,R2           ;INITIALIZE TABLE POINTER
                                 ;       VERIFY TAG BUS ACCORDING TO TABLE AND CYLINDER IN R3
 050404                          315$:

 050404  016037  000040  001142          MOV     RMMR2(R0),$BDDAT           ;STORE RMMR2 AT $BDDAT
 050412  042737  150000  001142          BIC     #RQA!RQB!TST,$BDDAT
 050420  011237  001140                  MOV     (R2),$GDDAT
 050424  050337  001140                  BIS     R3,$GDDAT          ;OR CYLINDER ADDRESS IN
 050430  023737  001140  001142          CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED AND RECEIVED
 050436  001407                          BEQ     320$               ;BRANCH IF TAG BUS OK
 050440  010037  001136                  MOV     R0,$BDADR
 050444  062737  000040  001136          ADD     #RMMR2,$BDADR
 050452  104266                          ERROR   266                ;INCORRECT TAG BUS DURING SEARCH
 050454  000420                          BR      340$
 050456                          320$:
                                 ;       ADVANCE TO NEXT ENTRY IN TABLE-EXIT IF DONE
 050456  062702  000002                  ADD     #2,R2
 050462  005712                          TST     (R2)
 050464  100407                          BMI     330$               ;EXIT IF ENTRY NEGATIVE
                                 ;       STEP THE COMMAND SEQUENCER AND REPEAT VERIFICATION

 050466  012760  141401  000024          MOV     #DMD!DBEN!MUR!MOC!DBCK,RMMR1(R0)          ;LOAD RMMR1

 050474  012760  041401  000024          MOV     #DMD!DBEN!MUR!MOC,RMMR1(R0)       ;LOAD RMMR1
 050502  000740                          BR      315$
 050504                          330$:
                                 ;REPEAT TAG BUS TEST FOR EACH PRIME CYLINDER, I.E., 1,2,4,...

 050504  006303                          ASL     R3                 ;SHIFT TO NEXT CYLINDER
 050506  020327  000512                  CMP     R3,#512
 050512  101001                          BHI     340$               ;EXIT IF 512 WAS DONE
 050514  000703                          BR      300$               ;TEST NEXT CYLINDER
 050516  000424                  340$:   BR      500$               ;JUMP OVER TABLE

 050520                          400$:

                                 ;TABLE OF TAG BUS CONTROL AND BIT VALUES

 050520  001777                          .WORD   1777               ;BUS BITS AT HIGH IMPEDANCE STATE
 050522  001777                          .WORD   1777
 050524  001777                          .WORD   1777
 050526  001777                          .WORD   1777
 050530  001777                          .WORD   1777
 050532  004000                          .WORD   CC       ;CONTROL BITS ENABLED, BIT 6 ON
 050534  004000                          .WORD   CC
 050536  024000                          .WORD   TAG!CC   ;TAG COMES ON
 050540  024000                          .WORD   TAG!CC
 050542  024000                          .WORD   TAG!CC
 050544  024000                          .WORD   TAG!CC
 050546  024000                          .WORD   TAG!CC
```

```
050550  024000                          .WORD   TAG!CC
050552  004000                          .WORD   CC      ;TAG GOES OFF
050554  004000                          .WORD   CC
050556  001777                          .WORD   1777            ;CONTROL BITS DISABLED
050560  001777                          .WORD   1777
050562  001777                          .WORD   1777
050564  001777                          .WORD   1777

050566  177777                          .WORD   -1              ;END OF TABLE

050570                          500$:                           ;END OF TEST

                                ;;*****************************************************************
                                ;*TEST 116      SEARCH TIMEOUT TEST

                                ;;*****************************************************************
050570  000004                  TST116: SCOPE
050572  012737  000116  001226          MOV     #116,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

050600  000240                          NOP
050602  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
050610  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
050616  012737  050632  001122          MOV     #T116,$LPADR    ;LOAD LOOP ON TEST ADDRESS
050624  012737  050632  001124          MOV     #T116,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
050632                          T116:
050632  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
050636  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
050642  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                ;SET VOLUME VALID USING SUBROUTINE
050646  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
050652  000402                          BR      10$             ;BRANCH TO 10$ IF NO ERROR
050654  104000                          ERROR                   ;RETURN HERE IF ERROR
050656  000550                          BR      90$
050660                          10$:
                                ;ENABLE DEBUG CLOCK AND LOAD SEARCH COMMAND

050660  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1

050666  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

050674  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

050702  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC

050710  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA

050716  012760  000031  000000          MOV     #SEARCH!GO,RMCS1(R0)    ;LOAD RMCS1
                                ;EXECUTE SEARCH TO TEST FOR ON LATCH RESET (17 CLOCKS)
050724  012702  000021                  MOV     #17.,R2
050730                          20$:

050730  012760  141401  000024          MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1

050736  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
050744  005302                          DEC     R2
050746  001370                          BNE     20$
```

```
                                        ;DROP ON CYLINDER TO RESET LATCH, RAISE ON CYLINDER
050750  012760  041001  000024                  MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
050756  012760  041401  000024                  MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
                                        ;STEP COMMAND SEQUENCER TO SECTOR COMPARE LOOP (2 CLOCKS)
050764  012702  000002                          MOV     #2,R2
050770                                  30$:
050770  012760  151401  000024                  MOV     #DMD!DBEN!MUR!MOC!DBCK!DTO,RMMR1(R0)    ;LOAD RMMR1
050776  012760  051401  000024                  MOV     #DMD!DBEN!MUR!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
051004  005302                                  DEC     R2
051006  001370                                  BNE     30$
                                        ;THE COMMAND SEQUENCER IS NOW IN THE SECTOR COMPARE LOOP.  STEP
                                        ;THROUGH THE LOOP AND VERIFY SEARCH IS ENABLED.
051010  012702  000005                          MOV     #5,R2
051014                                  40$:
051014  012760  151401  000024                  MOV     #DMD!DBEN!MUR!MOC!DBCK!DTO,RMMR1(R0)    ;LOAD RMMR1
051022  012760  051401  000024                  MOV     #DMD!DBEN!MUR!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
051030  016037  000024  001142                  MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
051036  042737  173777  001142                  BIC     #↑CESRC,$BDDAT
051044  001403                                  BEQ     50$             ;BRANCH IF SEARCH NOT ENABLED
051046  005302                                  DEC     R2
051050  001361                                  BNE     40$
051052  000412                                  BR      60$
051054  012737  004000  001140          50$:    MOV     #ESRC,$GDDAT    ;SETUP ERROR MSG
051062  010037  001136                          MOV     R0,$BDADR
051066  062737  000024  001136                  ADD     #RMMR1,$BDADR
051074  104265                                  ERROR   265             ;SEARCH NOT ENABLED
051076  000440                                  BR      90$
051100                                  60$:
                                        ;DROP DTO TO ENABLE SEARCH TIMEOUT AND WAIT FOR OPI TO SET.
051100  012760  041401  000024                  MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
051106  012737  000074  001524                  MOV     #60.,WATCH      ;SET WATCHDOG TIMER VALUE
051114  004777  130406                          JSR     PC,@CLOCK       ;START THE CLOCK
051120                                  70$:
051120  016037  000014  001142                  MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
051126  042737  157777  001142                  BIC     #↑COPI,$BDDAT
051134  001017                                  BNE     80$
051136  005737  001524                          TST     WATCH
051142  001366                                  BNE     70$
051144  004777  130360                          JSR     PC,@STOP        ;STOP THE CLOCK
051150  012737  020000  001140                  MOV     #OPI,$GDDAT     ;SETUP ERROR MSG
051156  010037  001136                          MOV     R0,$BDADR
051162  062737  000014  001136                  ADD     #RMER1,$BDADR
051170  104267                                  ERROR   267             ;OPI NOT SET BY SEARCH TIMEOUT
051172  000402                                  BR      90$
051174                                  80$:
051174  004777  130330                          JSR     PC,@STOP        ;STOP THE CLOCK
```

```
051200                              90$:                                    ;END OF TEST

                                    ;;*****************************************************************
                                    ;*TEST 117        DATA COMMAND TESTS (1)

                                    ;;*****************************************************************
051200  000004                      TST117: SCOPE
051202  012737  000117  001226              MOV     #117,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX

051210  000240                              NOP
051212  012737  000024  001120              MOV     #20.,$ICNT          ;20 ITERATIONS
051220  112737  000001  001131              MOVB    #1,$ERMAX           ;ONE ERROR ALLOWED
051226  012737  051242  001122              MOV     #T117,$LPADR    ;LOAD LOOP ON TEST ADDRESS
051234  012737  051242  001124              MOV     #T117,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
051242                              T117:
051242  012706  001100                      MOV     #STACK,SP           ;LOAD THE STACK POINTER
051246  013700  001276                      MOV     $BASE,R0            ;R0 = UNIBUS ADDRESS OF UUT
051252  013701  001456                      MOV     TSTQUE,R1           ;R1 = POINTER TO DEVICE
                                    ;;*****************************************************************
                                    ;VERIFY DATA COMMAND SETS OPI IF DRIVE NOT READY
                                    ;SET VOLUME VALID USING SUBROUTINE
051256  004737  060526                      JSR     PC,SETVV            ;GO SET VOLUME VALID
051262  000402                              BR      10$                 ;BRANCH TO 10$ IF NO ERROR
051264  104000                              ERROR                       ;RETURN HERE IF ERROR
051266  000471                              BR      40$
051270                              10$:
                                    ;       ENABLE DEBUG CLOCK AND LOAD READ COMMAND

051270  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1

051276  012760  000000  000014              MOV     #0,RMER1(R0)        ;LOAD RMER1

051304  012760  000000  000042              MOV     #0,RMER2(R0)        ;LOAD RMER2

051312  012760  000000  000006              MOV     #0,RMDA(R0)         ;LOAD RMDA

051320  012760  000000  000034              MOV     #0,RMDC(R0)         ;LOAD RMDC

051326  012760  000071  000000              MOV     #RD!GO,RMCS1(R0)        ;LOAD RMCS1
                                    ;       STEP COMMAND SEQUENCER TO UNIT READY TEST (3 CLOCKS)
051334  012702  000003                      MOV     #3,R2
051340                              20$:

051340  012760  141401  000024              MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

051346  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
051354  005302                              DEC     R2
051356  001370                              BNE     20$
                                    ;       DROP UNIT READY

051360  012760  040401  000024              MOV     #DMD!MOC!DBEN,RMMR1(R0) ;LOAD RMMR1
                                    ;       STEP SEQUENCER AND VERIFY OPI SETS (2 CLOCKS)
051366  012702  000002                      MOV     #2,R2
051372                              30$:

051372  012760  140401  000024              MOV     #DMD!MOC!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
```

```
051400  012760  040401  000024            MOV     #DMD!MOC!DBEN,RMMR1(R0) ;LOAD RMMR1
051406  005302                            DEC     R2
051410  001370                            BNE     30$
051412  016037  000014  001142            MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
051420  042737  157777  001142            BIC     #↑COPI,$BDDAT
051426  001011                            BNE     40$
051430  012737  020000  001140            MOV     #OPI,$GDDAT
051436  010037  001136                    MOV     R0,$BDADR
051442  062737  000014  001136            ADD     #RMER1,$BDADR
051450  104270                            ERROR   270             ;OPI NOT SET DURING DATA
051452  012737  051460  001124    40$:    MOV     #50$,$LPERR     ;CHANGE LOOP ON ERROR TEST
                                  ;;***********************************************************
                                  ;VERIFY DATA COMMAND ABORTS AT LOCATION 129
051460                            50$:
                                  ;SET VOLUME VALID USING SUBROUTINE
051460  004737  060526                    JSR     PC,SETVV        ;GO SET VOLUME VALID
051464  000402                            BR      60$             ;BRANCH TO 60$ IF NO ERROR
051466  104000                            ERROR                   ;RETURN HERE IF ERROR
051470  000576                            BR      150$
051472                            60$:
                                  ;ENABLE DEBUG CLOCK AND LOAD READ COMMAND
051472  012760  041401  000024            MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
051500  012760  000000  000014            MOV     #0,RMER1(R0)    ;LOAD RMER1
051506  012760  000000  000042            MOV     #0,RMER2(R0)    ;LOAD RMER2
051514  012760  000000  000006            MOV     #0,RMDA(R0)     ;LOAD RMDA
051522  012760  000000  000034            MOV     #0,RMDC(R0)     ;LOAD RMDC
051530  012760  000071  000000            MOV     #RD!GO,RMCS1(R0)        ;LOAD RMCS1
                                  ;STEP COMMAND SEQUENCER TO ABORT TEST AT LOCATION 129 (4 CLOCKS)
051536  012702  000004                    MOV     #4,R2
051542                            70$:
051542  012760  141401  000024            MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1
051550  012760  041401  000024            MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
051556  005302                            DEC     R2
051560  001370                            BNE     70$
                                  ;SET DEVICE FAULT TO CAUSE ABORT CONDITION
051562  012760  041501  000024            MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
                                  ;STEP THE SEQUENCER THROUGH THE TEST FOR ABORT (1 CLOCK)
051570  012702  000001                    MOV     #1,R2
051574                            80$:
051574  012760  141501  000024            MOV     #DMD!MUR!MOC!DBEN!MDF!DBCK,RMMR1(R0)    ;LOAD RMMR1
```

```
051602  012760  041501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
051610  005302                          DEC     R2
051612  001370                          BNE     80$

                                ;ABORT EBL SHOULD NOW BE ACTIVE - USE THE MAINTENANCE REGISTER TO
                                ;FORCE BIT CLOCKS AND VERIFY THAT EBL SETS WITHIN 16 BIT CLOCKS
051614  012702  000020                  MOV     #16.,R2         ;MAXIMUM NUMBER OF BIT CLOCKS
051620                          85$:

051620  012760  045501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF!MCLK,RMMR1(R0)    ;LOAD RMMR1

051626  012760  041501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1

051634  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
051642  042737  157777  001142          BIC     #↑CEBL,$BDDAT
051650  001014                          BNE     90$             ;BRANCH IF EBL IS SET

051652  005302                          DEC     R2
051654  001361                          BNE     85$             ;CONTINUE BIT CLOCKS IF COUNT NOT 0
051656  012737  020000  001140          MOV     #EBL,$GDDAT
051664  010037  001136                  MOV     R0,$BDADR
051670  062737  000024  001136          ADD     #RMMR1,$BDADR
051676  104271                          ERROR   271             ;EBL NOT SET AT DATA ABORT
051700  000472                          BR      150$
051702                          90$:
                                ;STEP THE SEQUENCER THROUGH ITS TEST FOR EBL (2 CLOCKS)
051702  012702  000002                  MOV     #2,R2
051706                          100$:

051706  012760  141501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF!DBCK,RMMR1(R0)    ;LOAD RMMR1

051714  012760  041501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
051722  005302                          DEC     R2
051724  001370                          BNE     100$

                                ;ABORT EBL SHOULD NOW BE INACTIVE - FORCE BIT CLOCK USING THE
                                ;MAINTENANCE REGISTER TO RESET EBL (16 BIT CLOCKS)
051726  012702  000020                  MOV     #16.,R2         ;MAXIMUM NUMBER OF BIT CLOCKS
051732                          110$:

051732  012760  045501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF!MCLK,RMMR1(R0)    ;LOAD RMMR1

051740  012760  041501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
051746  005302                          DEC     R2
051750  001370                          BNE     110$            ;ISSUE 16 BIT CLOCKS THEN TEST
051752                          120$:
                                ;VERIFY EBL IS NOW RESET

051752  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
051760  042737  157777  001142          BIC     #↑CEBL,$BDDAT
051766  001411                          BEQ     130$            ;BRANCH IF EBL IS RESET
051770  005037  001140                  CLR     $GDDAT
051774  010037  001136                  MOV     R0,$BDADR
052000  062737  000024  001136          ADD     #RMMR1,$BDADR
052006  104273                          ERROR   273             ;EBL DIDNT RESET ON TIME
052010  000426                          BR      150$
```

```
052012                                 130$:
                                       ;VERIFY GO RESETS WITHIN 4 CLOCK CYCLES
052012  012702  000004                         MOV     #4,R2
052016                                 140$:
052016  012760  141501  000024                 MOV     #DMD!MUR!MOC!DBEN!MDF!DBCK,RMMR1(R0)      ;LOAD RMMR1
052024  012760  041501  000024                 MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
052032  005302                                 DEC     R2
052034  001370                                 BNE     140$
052036  016037  000000  001142                 MOV     RMCS1(R0),$BDDAT          ;STORE RMCS1 AT $BDDAT
052044  042737  177776  001142                 BIC     #↑CGO,$BDDAT
052052  001405                                 BEQ     150$
052054  005037  001140                         CLR     $GDDAT
052060  010037  001136                         MOV     R0,$BDADR
052064  104274                                 ERROR   274             ;GO NOT RESET DURING DATA ABORT
052066  012737  052074  001124         150$:   MOV     #200$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS
                                       ;;**************************************************************
                                       ;VERIFY SEQUENCER BRANCHES TO SEEK WHEN RUN AND GO FLOP SETS
052074                                 200$:
                                       ;SET VOLUME VALID USING SUBROUTINE
052074  004737  060526                         JSR     PC,SETVV        ;GO SET VOLUME VALID
052100  000402                                 BR      210$            ;BRANCH TO 210$ IF NO ERROR
052102  104000                                 ERROR                   ;RETURN HERE IF ERROR
052104  000512                                 BR      250$
052106                                 210$:
                                       ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND
052106  012760  041401  000024                 MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)      ;LOAD RMMR1
052114  012760  000000  000014                 MOV     #0,RMER1(R0)    ;LOAD RMER1
052122  012760  000000  000042                 MOV     #0,RMER2(R0)    ;LOAD RMER2
052130  012760  000000  000006                 MOV     #0,RMDA(R0)     ;LOAD RMDA
052136  012760  000000  000034                 MOV     #0,RMDC(R0)     ;LOAD RMDC
052144  012760  000071  000000                 MOV     #RD!GO,RMCS1(R0)        ;LOAD RMCS1
                                       ;MOVE SEQUENCER TO TEST FOR RUN AND GO AT LOCATION 130 (5 CLOCKS)
052152  012702  000005                         MOV     #5,R2
052156                                 220$:
052156  012760  141401  000024                 MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)         ;LOAD RMMR1
052164  012760  041401  000024                 MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)      ;LOAD RMMR1
052172  005302                                 DEC     R2
052174  001370                                 BNE     220$
                                       ;VERIFY RUN AND GO IS SET
052176  016037  000024  001142                 MOV     RMMR1(R0),$BDDAT          ;STORE RMMR1 AT $BDDAT
052204  042737  137777  001142                 BIC     #↑CRG,$BDDAT
```

K04


```
052212  001012                          BNE     230$
052214  012737  040000  001140          MOV     #RG,$GDDAT
052222  010037  001136                  MOV     R0,$BDADR
052226  062737  000024  001136          ADD     #RMMR1,$BDADR
052234  104275                          ERROR   275             ;RUN AND GO NOT SET
052236  000435                          BR      250$
052240                          230$:
                                ;VERIFY THAT CYLINDER TAG COMES UP IN ONE CLOCK CYCLE
052240  012702  000001                  MOV     #1,R2
052244                          240$:

052244  012760  141401  000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

052252  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
052260  005302                          DEC     R2
052262  001370                          BNE     240$

052264  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
052272  042737  160000  001142          BIC     #RQA!RQB!TAG,$BDDAT
052300  012737  004000  001140          MOV     #CC,$GDDAT
052306  023737  001140  001142          CMP     $GDDAT,$BDDAT
052314  001406                          BEQ     250$
052316  010037  001136                  MOV     R0,$BDADR
052322  062737  000040  001136          ADD     #RMMR2,$BDADR
052330  104276                          ERROR   276             ;CYLINDER TAG NOT SET DURING DATA

052332  012737  052344  001124  250$:   MOV     #260$,$LPERR    ;CHANGE LOOP ON ERROR ADDRESS

                                ;;************************************************************
                                ;VERIFY DATA COMMAND ABORTS AT COMMAND SEQUENCER LOCATIONS 144, 145
052340  012702  000144                  MOV     #144,R2         ;INITIALIZE TEST LOCATION
052344                          260$:
                                ;SET VOLUME VALID USING SUBROUTINE
052344  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
052350  000402                          BR      270$            ;BRANCH TO 270$ IF NO ERROR
052352  104000                          ERROR                   ;RETURN HERE IF ERROR
052354  000553                          BR      320$
052356                          270$:
                                ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND

052356  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1

052364  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

052372  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

052400  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA

052406  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC

052414  012760  000071  000000          MOV     #RD!GO,RMCS1(R0)        ;LOAD RMCS1
                                ;WAIT FOR RUN AND GO TO SET
052422  012737  000310  001524          MOV     #200.,WATCH     ;SET WATCHDOG TIMER VALUE
052430  004777  127072                  JSR     PC,@CLOCK       ;START THE CLOCK
052434                          280$:
```

```
052434 016037 000024 001142          MOV     RMMR1(R0),$BDDAT         ;STORE RMMR1 AT $BDDAT
052442 042737 137777 001142          BIC     #↑CRG,$BDDAT
052450 001017                        BNE     290$
052452 005737 001524                 TST     WATCH
052456 001366                        BNE     280$
052460 004777 127044                 JSR     PC,@STOP        ;STOP THE CLOCK
052464 012737 040000 001140          MOV     #RG,$GDDAT
052472 010037 001136                 MOV     R0,$BDADR
052476 062737 000024 001136          ADD     #RMMR1,$BDADR
052504 104275                        ERROR   275             ;RUN AND GO NOT SET
052506 000476                        BR      320$
052510                       290$:
052510 004777 127014                 JSR     PC,@STOP        ;STOP THE CLOCK
                             ;MOVE COMMAND SEQUENCER TO ABORT TEST (LOCATION 144 OR 145)
052514 012703 000006                 MOV     #6,R3           ;SETUP CLOCK COUNT
052520 022702 000144                 CMP     #144,R2
052524 001402                        BEQ     300$
052526 012703 000007                 MOV     #7,R3
052532                       300$:

052532 012760 141401 000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)           ;LOAD RMMR1

052540 012760 041401 000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)      ;LOAD RMMR1
052546 005303                        DEC     R3
052550 001370                        BNE     300$
                             ;SET DRIVE FAULT TO FORCE ABORT CONDITION

052552 012760 041501 000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
                             ;CLOCK SEQUENCER THROUGH ITS TEST FOR ABORT (1 CLOCK)
052560 012703 000001                 MOV     #1,R3
052564                       305$:

052564 012760 141501 000024          MOV     #DMD!MUR!MOC!DBEN!MDF!DBCK,RMMR1(R0)       ;LOAD RMMR1

052572 012760 041501 000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
052600 005303                        DEC     R3
052602 001370                        BNE     305$            ;ISSUE 2 CLOCKS

                             ;ABORT EBL SHOULD NOW BE ACTIVE - USE THE MAINTENANCE REGISTER TO
                             ;FORCE BIT CLOCKS AND VERIFY THAT EBL SETS WITHIN 16 BIT CLOCKS
052604 012702 000020                 MOV     #16.,R2         ;MAXIMUM NUMBER OF BIT CLOCKS
052610                       306$:

052610 012760 045501 000024          MOV     #DMD!MUR!MOC!DBEN!MDF!MCLK,RMMR1(R0)       ;LOAD RMMR1

052616 012760 041501 000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1

052624 016037 000024 001142          MOV     RMMR1(R0),$BDDAT         ;STORE RMMR1 AT $BDDAT
052632 042737 157777 001142          BIC     #↑CEBL,$BDDAT
052640 001013                        BNE     310$            ;BRANCH IF EBL IS SET

052642 005302                        DEC     R2
052644 001361                        BNE     306$            ;CONTINUE BIT CLOCKS IF COUNT NOT 0
052646 012737 020000 001140          MOV     #EBL,$GDDAT
052654 010037 001136                 MOV     R0,$BDADR
052660 062737 000024 001136          ADD     #RMMR1,$BDADR
```

```
052666  104271                          ERROR   271                     ;SEQ DID NOT ABORT AT DATA SEEK
052670  022702  000144          310$:   CMP     #144,R2
052674  001003                          BNE     320$
052676  012702  000145                  MOV     #145,R2
052702  000620                          BR      260$
052704  012737  052716  001124  320$:   MOV     #330$,$LPERR                    ;CHANGE LOOP ON ERROR ADDRESS
                                ;;*****************************************************************
                                ;VERIFY HEAD TAG DURING DATA COMMAND
052712  012702  000400                  MOV     #400,R2                 ;INITIALIZE TRACK ADDRESS
052716                          330$:
                                ;SET VOLUME VALID USING SUBROUTINE
052716  004737  060526                  JSR     PC,SETVV                ;GO SET VOLUME VALID
052722  000402                          BR      340$                    ;BRANCH TO 340$ IF NO ERROR
052724  104000                          ERROR                           ;RETURN HERE IF ERROR
052726  000550                          BR      400$
052730                          340$:
                                ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND
052730  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
052736  012760  000000  000014          MOV     #0,RMER1(R0)            ;LOAD RMER1
052744  012760  000000  000042          MOV     #0,RMER2(R0)            ;LOAD RMER2
052752  010260  000006                  MOV     R2,RMDA(R0)             ;LOAD RMDA
052756  012760  000000  000034          MOV     #0,RMDC(R0)             ;LOAD RMDC
052764  012760  000071  000000          MOV     #RD!GO,RMCS1(R0)                ;LOAD RMCS1
                                ;WAIT FOR RUN AND GO TO SET
052772  012737  000310  001524          MOV     #200.,WATCH             ;SET WATCHDOG TIMER VALUE
053000  004777  126522                  JSR     PC,@CLOCK               ;START THE CLOCK
053004                          350$:
053004  016037  000024  001142          MOV     RMMR1(R0),$BDDAT                ;STORE RMMR1 AT $BDDAT
053012  042737  137777  001142          BIC     #^CRG,$BDDAT
053020  001017                          BNE     360$
053022  005737  001524                  TST     WATCH
053026  001366                          BNE     350$
053030  004777  126474                  JSR     PC,@STOP                ;STOP THE CLOCK
053034  012737  040000  001140          MOV     #RG,$GDDAT
053042  010037  001136                  MOV     R0,$BDADR
053046  062737  000024  001136          ADD     #RMMR1,$BDADR
053054  104275                          ERROR   275                     ;RUN AND GO NOT SET
053056  000474                          BR      400$
053060                          360$:
053060  004777  126444                  JSR     PC,@STOP                ;STOP THE CLOCK
                                ;STEP COMMAND SEQUENCER TO HEAD SEQUENCE, LOCATION 156 (17 CLOCKS)
053064  012703  000021                  MOV     #17.,R3
053070                          370$:
053070  012760  141401  000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1
053076  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
```

```
053104  005303                              DEC     R3
053106  001370                              BNE     370$
                                    ;DROP AND RAISE ON CYLINDER TO RESET ON LATCH

053110  012760  041001  000024              MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

053116  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
053124  012703  053252                      MOV     #450$,R3        ;INITIALIZE TABLE POINTER
                                    ;VERIFY TAG BUS ACCORDING TO TABLE AND TRACK ADDRESS IN R2
053130                              375$:

053130  016037  000040  001142              MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
053136  042737  150000  001142              BIC     #RQA!RQB!TST,$BDDAT
053144  011337  001140                      MOV     (R3),$GDDAT
053150  010204                              MOV     R2,R4           ;GENERATE EXPECTED TAG BUS
053152  000304                              SWAB    R4
053154  042704  177770                      BIC     #^C7,R4
053160  050437  001140                      BIS     R4,$GDDAT
053164  023737  001140  001142              CMP     $GDDAT,$BDDAT
053172  001020                              BNE     390$

053174  012760  141401  000024              MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

053202  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
                                    ;ADVANCE TO NEXT TABLE ENTRY
053210  062703  000002                      ADD     #2,R3
053214  005713                              TST     (R3)
053216  100401                              BMI     380$
053220  000743                              BR      375$
053222                              380$:
                                    ;SHIFT TO NEXT TRACK ADDRESS-EXIT LOOP IF DONE
053222  006302                              ASL     R2
053224  020227  002000                      CMP     R2,#2000
053230  101007                              BHI     400$
053232  000631                              BR      330$
053234                              390$:
                                    ;ERROR ON TAG BUS DURING HEAD SEQUENCE
053234  010037  001136                      MOV     R0,$BDADR
053240  062737  000040  001136              ADD     #RMMR2,$BDADR
053246  104276                              ERROR   276             ;INCORRECT TAG BUS

053250  000440                      400$:   BR      500$            ;JUMP OVER TABLE

053252                              450$:
                                    ;TABLE OF TAG BUS DURING HEAD SEQUENCE
053252  002000                              .WORD   CH
053254  002000                              .WORD   CH
053256  022000                              .WORD   CH!TAG
053260  022000                              .WORD   CH!TAG
053262  022000                              .WORD   CH!TAG
053264  022000                              .WORD   CH!TAG
053266  022000                              .WORD   CH!TAG
053270  022000                              .WORD   CH!TAG
053272  002000                              .WORD   CH
053274  002000                              .WORD   CH
053276  001777                              .WORD   1777
```

```
053300  001777                          .WORD   1777
053302  001777                          .WORD   1777
053304  001777                          .WORD   1777
053306  001777                          .WORD   1777
053310  001777                          .WORD   1777
053312  001777                          .WORD   1777
053314  001777                          .WORD   1777
053316  001777                          .WORD   1777
053320  001777                          .WORD   1777
053322  001777                          .WORD   1777
053324  001777                          .WORD   1777
053326  001777                          .WORD   1777
053330  001777                          .WORD   1777
053332  001777                          .WORD   1777
053334  001777                          .WORD   1777
053336  001777                          .WORD   1777
053340  001777                          .WORD   1777
053342  001777                          .WORD   1777
053344  001777                          .WORD   1777
053346  001777                          .WORD   1777

053350  177777                          .WORD   -1              ;END OF TABLE

053352                          500$:                           ;END OF TEST


                                ;;*****************************************************************
                                ;*TEST 120      DATA COMMAND TESTS (2)

                                ;;*****************************************************************
053352  000004                  TST120: SCOPE
053354  012737  000120  001226          MOV     #120,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

053362  000240                          NOP
053364  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
053372  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
053400  012737  053414  001122          MOV     #T120,$LPADR    ;LOAD LOOP ON TEST ADDRESS
053406  012737  053414  001124          MOV     #T120,$LPERR    ;LOAD LOOP ON ERROR ADDRESS
053414                          T120:
053414  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
053420  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
053424  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                ;;*****************************************************************
                                ;VERIFY OPI SETS IF ON CYLINDER LATCH DOESNT RESET
                                ;SET VOLUME VALID USING SUBROUTINE
053430  004737  060526                  JSR     PC,SETVV        ;GO SET VOLUME VALID
053434  000402                          BR      10$             ;BRANCH TO 10$ IF NO ERROR
053436  104000                          ERROR                   ;RETURN HERE IF ERROR
053440  000514                          BR      60$
053442                          10$:
                                ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND DURING CYLINDER SEQUENCE

053442  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1

053450  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

053456  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
```

```

053464 012760 000000 000006          MOV     #0,RMDA(R0)        ;LOAD RMDA

053472 012760 000000 000034          MOV     #0,RMDC(R0)        ;LOAD RMDC

053500 012760 000071 000000          MOV     #RD!GO,RMCS1(R0)           ;LOAD RMCS1
                              ;WAIT FOR RUN AND GO TO SET
053506 012737 000310 001524          MOV     #200.,WATCH        ;SET WATCHDOG TIMER VALUE
053514 004777 126006                 JSR     PC,@CLOCK          ;START THE CLOCK
053520                        20$:

053520 016037 000024 001142          MOV     RMMR1(R0),$BDDAT           ;STORE RMMR1 AT $BDDAT
053526 042737 137777 001142          BIC     #↑CRG,$BDDAT
053534 001017                        BNE     30$
053536 005737 001524                 TST     WATCH
053542 001366                        BNE     20$
053544 004777 125760                 JSR     PC,@STOP           ;STOP THE CLOCK
053550 012737 040000 001140          MOV     #RG,$GDDAT
053556 010037 001136                 MOV     R0,$BDADR
053562 062737 000024 001136          ADD     #RMMR1,$BDADR
053570 104275                        ERROR   275                ;RUN AND GO DIDNT SET
053572 000437                        BR      60$
053574                        30$:
053574 004777 125730                 JSR     PC,@STOP           ;STOP THE CLOCK
                              ;STEP COMMAND SEQUENCER AND VERIFY OPI SETS (19 CLOCKS)
053600 012702 000023                 MOV     #19.,R2
053604                        40$:

053604 012760 141401 000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)           ;LOAD RMMR1

053612 012760 041401 000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)        ;LOAD RMMR1
053620 005302                        DEC     R2
053622 001370                        BNE     40$

053624 016037 000014 001142          MOV     RMER1(R0),$BDDAT           ;STORE RMER1 AT $BDDAT
053632 042737 157777 001142          BIC     #↑COPI,$BDDAT
053640 001011                        BNE     50$                ;BRANCH IF OPI SET
053642 012737 020000 001140          MOV     #OPE,$GDDAT
053650 010037 001136                 MOV     R0,$BDADR
053654 062737 000014 001136          ADD     #RMER1,$BDADR
053662 104277                        ERROR   277                ;OPI NOT SET BY ON LATCH SET
053664 012737 053672 001124   50$:   MOV     #60$,$LPERR        ;CHANGE LOOP ON ERROR ADDRESS
                              ;;**********************************************************
                              ;VERIFY DATA COMMAND ABORTS DURING SEEK WAIT LOOP
053672                        60$:
                              ;SET VOLUME VALID USING SUBROUTINE
053672 004737 060526                 JSR     PC,SETVV           ;GO SET VOLUME VALID
053676 000402                        BR      70$                ;BRANCH TO 70$ IF NO ERROR
053700 104000                        ERROR                      ;RETURN HERE IF ERROR
053702 000567                        BR      140$
053704                        70$:
                              ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND

053704 012760 041401 000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)        ;LOAD RMMR1

053712 012760 000000 000014          MOV     #0,RMER1(R0)       ;LOAD RMER1
```

```
053720  012760  000000  000042          MOV     #0,RMER2(R0)        ;LOAD RMER2
053726  012760  000000  000034          MOV     #0,RMDC(R0)         ;LOAD RMDC
053734  012760  000000  000006          MOV     #0,RMDA(R0)         ;LOAD RMDA
053742  012760  000071  000000          MOV     #RD!GO,RMCS1(R0)            ;LOAD RMCS1
                                ;WAIT FOR RUN & GO TO SET
053750  012737  000310  001524          MOV     #200.,WATCH         ;SET WATCHDOG TIMER VALUE
053756  004777  125544                  JSR     PC,@CLOCK           ;START THE CLOCK
053762                          80$:
053762  016037  000024  001142          MOV     RMMR1(R0),$BDDAT            ;STORE RMMR1 AT $BDDAT
053770  042737  137777  001142          BIC     #^CRG,$BDDAT
053776  001017                          BNE     90$
054000  005737  001524                  TST     WATCH
054004  001366                          BNE     80$
054006  004777  125516                  JSR     PC,@STOP            ;STOP THE CLOCK
054012  012737  040000  001140          MOV     #RG,$GDDAT
054020  010037  001136                  MOV     R0,$BDADR
054024  062737  000024  001136          ADD     #RMMR1,$BDADR
054032  104275                          ERROR   275                 ;RUN AND GO DIDNT SET
054034  000512                          BR      140$
054036                          90$:
054036  004777  125466                  JSR     PC,@STOP            ;STOP THE CLOCK
                                ;STEP COMMAND SEQUENCER TO ON LATCH TEST AT LOCATION 156 (17 CLOCKS)
054042  012702  000021                  MOV     #17.,R2
054046                          100$:
054046  012760  141401  000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)            ;LOAD RMMR1
054054  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)         ;LOAD RMMR1
054062  005302                          DEC     R2
054064  001370                          BNE     100$
                                ;DROP ON CYLINDER TO RESET LATCH
054066  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                ;MOVE COMMAND SEQUENCER TO SEEK WATI LOOP (31 CLOCKS)
054074  012702  000037                  MOV     #31.,R2
054100                          110$:
054100  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1
054106  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
054114  005302                          DEC     R2
054116  001370                          BNE     110$
                                ;STEP THROUGH SEEK WAIT LOOP (6 CLOCKS) 2 TIMES
054120  012702  000006                  MOV     #6.,R2
054124                          120$:
054124  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1
054132  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
054140  005302                          DEC     R2
054142  001370                          BNE     120$
```

```
                                        ;SET SEEK INCOMPLETE ERROR TO CAUSE ABORT
054144  012760  041201  000024                 MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)      ;LOAD RMMR1
                                        ;CLOCK THE SEQUENCER THROUGH ITS TEST FOR ABORT (2 CLOCKS)
054152  012702  000002                         MOV     #2,R2
054156                                  130$:
054156  012760  141201  000024                 MOV     #DMD!MUR!DBEN!MSER!DBCK,RMMR1(R0)        ;LOAD RMMR1
054164  012760  041201  000024                 MOV     #DMD!MUR!DBEN!MSER,RMMR1(R0)      ;LOAD RMMR1
054172  005302                                 DEC     R2
054174  001370                                 BNE     130$
                                        ;ABORT EBL SHOULD NOW BE ACTIVE - USE THE MAINTENANCE REGISTER TO
                                        ;FORCE BIT CLOCKS AND VERIFY THAT EBL SETS WITHIN 16 BIT CLOCKS
054176  012702  000020                         MOV     #16.,R2            ;MAXIMUM NUMBER OF BIT CLOCKS
054202                                  135$:
054202  012760  045501  000024                 MOV     #DMD!MUR!MOC!DBEN!MDF!MCLK,RMMR1(R0)     ;LOAD RMMR1
054210  012760  041501  000024                 MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
054216  016037  000024  001142                 MOV     RMMR1(R0),$BDDAT             ;STORE RMMR1 AT $BDDAT
054224  042737  157777  001142                 BIC     #^CEBL,$BDDAT
054232  001013                                 BNE     140$
054234  005302                                 DEC     R2
054236  001361                                 BNE     135$               ;CONTINUE BIT CLOCKS IF COUNT NOT 0
054240  012737  020000  001140                 MOV     #EBL,$GDDAT
054246  010037  001136                         MOV     R0,$BDADR
054252  062737  000024  001136                 ADD     #RMMR1,$BDADR
054260  104300                                 ERROR   300                ;EBL NOT SET DUE TO ABORT
054262  012737  054270  001124          140$:  MOV     #150$,$LPERR                 ;CHANGE LOOP ON ERROR ADDRESS
                                        ;;*************************************************************
                                        ;VERIFY DATA COMMAND ABORTS DURING OFFSET IF ON CYLINDER LATCH
                                        ;DOESNT RESET
054270                                  150$:
                                        ;SET VOLUME VALID USING SUBROUTINE
054270  004737  060526                         JSR     PC,SETVV           ;GO SET VOLUME VALID
054274  000402                                 BR      160$               ;BRANCH TO 160$ IF NO ERROR
054276  104000                                 ERROR                      ;RETURN HERE IF ERROR
054300  000536                                 BR      220$
054302                                  160$:
                                        ;LOAD TRACK,SETCOR,CYLINDER ADDRESS
054302  012760  000000  000006                 MOV     #0,RMDA(R0)        ;LOAD RMDA
054310  012760  000000  000034                 MOV     #0,RMDC(R0)        ;LOAD RMDC
                                        ;SET OFFSET MODE USING SUBROUTINE
054316  004737  060656                         JSR     PC,SETOM           ;GO SET OFFSET MODE
054322  000402                                 BR      170$               ;BRANCH TO 170$ IF NO ERROR
054324  104000                                 ERROR                      ;RETURN HERE IF ERROR
054326  000523                                 BR      220$
054330                                  170$:
                                        ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND
```

```
054330  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1

054336  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1

054344  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2

054352  012760  000071  000000          MOV     #RD!GO,RMCS1(R0)        ;LOAD RMCS1
                                ;WAIT FOR RUN AND GO TO SET
054360  012737  000310  001524          MOV     #200.,WATCH     ;SET WATCHDOG TIMER VALUE
054366  004777  125134                  JSR     PC,@CLOCK       ;START THE CLOCK
054372                          180$:

054372  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
054400  042737  137777  001142          BIC     #^CRG,$BDDAT
054406  001017                          BNE     190$
054410  005737  001524                  TST     WATCH
054414  001366                          BNE     180$
054416  004777  125106                  JSR     PC,@STOP        ;STOP THE CLOCK
054422  012737  040000  001140          MOV     #RG,$GDDAT
054430  010037  001140                  MOV     R0,$GDDAT
054434  062737  000024  001136          ADD     #RMMR1,$BDADR
054442  104275                          ERROR   275             ;RUN AND GO NOT SET
054444  000454                          BR      220$
054446                          190$:
054446  004777  125056                  JSR     PC,@STOP        ;STOP THE CLOCK
                                ;STEP COMMAND SEQUENCER TO ON LATCH TEST AT LOCATION 156 (17 CLOCKS)
054452  012702  000021                  MOV     #17.,R2
054456                          200$:

054456  012760  141401  000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

054464  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
054472  005302                          DEC     R2
054474  001370                          BNE     200$
                                ;DROP ON CYLINDER TO RESET LATCH, SET ON CYLINDER TO PASS TEST
                                ;AT LOCATION 166

054476  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

054504  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
                                ;MOVE SEQUENCER TO SET OPI AND EBL (39 CLOCKS)
054512  012702  000047                  MOV     #39.,R2
054516                          210$:

054516  012760  141401  000024          MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1

054524  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
054532  005302                          DEC     R2
054534  001370                          BNE     210$
                                ;VERIFY OPI IS SET

054536  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
054544  042737  157777  001142          BIC     #^COPI,$BDDAT
054552  001011                          BNE     220$
054554  012737  020000  001140          MOV     #OPI,$GDDAT
```

```
054562  010037  001136                      MOV     R0,$BDADR
054566  062737  000014  001136              ADD     #RMER1,$BDADR
054574  104277                              ERROR   277                     ;OPI NOT SET DURING OFFSET
054576  012737  054604  001124      220$:   MOV     #230$,$LPERR            ;CHANGE LOOP ON ERROR ADDRESS

                                    ;;*************************************************************
                                    ;VERIFY DATA COMMAND ABORTS DURING OFFSET WAIT LOOP

054604                              230$:
                                    ;SET VOLUME VALID USING SUBROUTINE
054604  004737  060526                      JSR     PC,SETVV                ;GO SET VOLUME VALID
054610  000403                              BR      240$                    ;BRANCH TO 240$ IF NO ERROR
054612  104000                              ERROR                           ;RETURN HERE IF ERROR
054614  000137  055224                      JMP     310$
054620                              240$:
                                    ;LOAD SECTOR,TRACK AND CYLINDER ADDRESS

054620  012760  000000  000006              MOV     #0,RMDA(R0)             ;LOAD RMDA

054626  012760  000000  000034              MOV     #0,RMDC(R0)             ;LOAD RMDC
                                    ;SET OFFSET MODE USING SUBROUTINE
054634  004737  060656                      JSR     PC,SETOM                ;GO SET OFFSET MODE
054640  000402                              BR      245$                    ;BRANCH TO 245$ IF NO ERROR
054642  104000                              ERROR                           ;RETURN HERE IF ERROR
054644  000567                              BR      310$
                                    ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND
054646                              245$:

054646  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1

054654  012760  000000  000014              MOV     #0,RMER1(R0)            ;LOAD RMER1

054662  012760  000000  000042              MOV     #0,RMER2(R0)            ;LOAD RMER2

054670  012760  000071  000000              MOV     #RD!GO,RMCS1(R0)                ;LOAD RMCS1
                                    ;WAIT FOR RUN AND GO TO SET
054676  012737  000310  001524              MOV     #200.,WATCH             ;SET WATCHDOG TIMER VALUE
054704  004777  124616                      JSR     PC,@CLOCK               ;START THE CLOCK
054710                              250$:

054710  016037  000024  001142              MOV     RMMR1(R0),$BDDAT                ;STORE RMMR1 AT $BDDAT
054716  042737  137777  001142              BIC     #^CRG,$BDDAT
054724  001017                              BNE     260$
054726  005737  001524                      TST     WATCH
054732  001366                              BNE     250$
054734  004777  124570                      JSR     PC,@STOP                ;STOP THE CLOCK
054740  012737  040000  001140              MOV     #RG,$GDDAT
054746  010037  001136                      MOV     R0,$BDADR
054752  062737  000024  001136              ADD     #RMMR1,$BDADR
054760  104275                              ERROR   275                     ;RUN AND GO NOT SET
054762  000520                              BR      310$
054764                              260$:
054764  004777  124540                      JSR     PC,@STOP                ;STOP THE CLOCK
                                    ;STEP SEQUENCER TO LOCATION 156 (17 CLOCKS)
054770  012702  000021                      MOV     #17.,R2
054774                              270$:
```

```
055774  012760  141401  000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1
055002  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
055010  005302                          DEC     R2
055012  001370                          BNE     270$
                                ;DROP ON CYLINDER TO RESET LATCH, SET ON CYLINDER TO PASS TEST
                                ;AT LOCATION 166
055014  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
055022  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
                                ;MOVE SEQUENCER TO LOCATION  174 (37 CLOCKS)
055030  012702  000045                  MOV     #37.,R2
055034                          280$:
055034  012760  141401  000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1
055042  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
055050  005302                          DEC     R2
055052  001370                          BNE     280$
                                ;DROP ON CYLINDER TO RESET LATCH, LEAVE ON CYLINDER RESET
055054  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                ;STEP SEQUENCER THROUGH OFFSET WAIT LOOP TWICE (7 CLOCKS)
055062  012702  000007                  MOV     #7.,R2
055066                          290$:
055066  012760  141001  000024          MOV     #DMD!MUR!DBEN!DBCK,RMMR1(R0)    ;LOAD RMMR1
055074  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
055102  005302                          DEC     R2
055104  001370                          BNE     290$
                                ;SET DRIVE FAULT TO CAUSE ABORT CONDITION
055106  012760  041101  000024          MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
                                ;STEP SEQUENCER THROUGH ITS TEST FOR ABORT (2 CLOCKS)
055114  012702  000002                  MOV     #2,R2
055120                          300$:
055120  012760  141101  000024          MOV     #DMD!MUR!DBEN!MDF!DBCK,RMMR1(R0)        ;LOAD RMMR1
055126  012760  041101  000024          MOV     #DMD!MUR!DBEN!MDF,RMMR1(R0)     ;LOAD RMMR1
055134  005302                          DEC     R2
055136  001370                          BNE     300$
                                ;ABORT EBL SHOULD NOW BE ACTIVE - USE THE MAINTENANCE REGISTER TO
                                ;FORCE BIT CLOCKS AND VERIFY THAT EBL SETS WITHIN 16 BIT CLOCKS
055140  012702  000020                  MOV     #16.,R2         ;MAXIMUM NUMBER OF BIT CLOCKS
055144                          305$:
055144  012760  045501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF!MCLK,RMMR1(R0)    ;LOAD RMMR1
055152  012760  041501  000024          MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
055160  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
```

```
055166  042737  157777  001142          BIC     #↑CEBL,$BDDAT
055174  001013                          BNE     310$

055176  005302                          DEC     R2
055200  001361                          BNE     305$                    ;CONTINUE BIT CLOCKS IF COUNT NOT 0
055202  012737  020000  001140          MOV     #EBL,$GDDAT
055210  010037  001136                  MOV     R0,$BDADR
055214  062737  000024  001136          ADD     #RMMR1,$BDADR
055222  104271                          ERROR   271                     ;EBL NOT SET DUE TO OFFSET WAIT ABORT

055224  012737  055232  001124  310$:   MOV     #320$,$LPERR            ;CHANGE LOOP ON ERROR ADDRESS

                                ;;**************************************************************
                                ;VERIFY THAT DATA COMMAND ABORTS DURING SECTOR WAIT LOOP AT LOCATION 179

055232                          320$:
                                ;SET VOLUME VALID USING SUBROUTINE
055232  004737  060526                  JSR     PC,SETVV                ;GO SET VOLUME VALID
055236  000403                          BR      330$                    ;BRANCH TO 330$ IF NO ERROR
055240  104000                          ERROR                           ;RETURN HERE IF ERROR
055242  000137  055670                  JMP     420$
055246                          330$:
                                ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND

055246  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1

055254  012760  000000  000014          MOV     #0,RMER1(R0)            ;LOAD RMER1

055262  012760  000000  000042          MOV     #0,RMER2(R0)            ;LOAD RMER2

055270  012760  000000  000006          MOV     #0,RMDA(R0)             ;LOAD RMDA

055276  012760  000000  000034          MOV     #0,RMDC(R0)             ;LOAD RMDC

055304  012760  000071  000000          MOV     #RD!GO,RMCS1(R0)                ;LOAD RMCS1
                                ;WAIT FOR RUN AND GO TO SET
055312  012737  000310  001524          MOV     #200.,WATCH             ;SET WATCHDOG TIMER VALUE
055320  004777  124202                  JSR     PC,@CLOCK               ;START THE CLOCK
055324                          340$:

055324  016037  000024  001142          MOV     RMMR1(R0),$BDDAT                ;STORE RMMR1 AT $BDDAT
055332  042737  137777  001142          BIC     #↑CRG,$BDDAT
055340  001017                          BNE     350$
055342  005737  001524                  TST     WATCH
055346  001366                          BNE     340$
055350  004777  124154                  JSR     PC,@STOP                ;STOP THE CLOCK
055354  012737  040000  001140          MOV     #RG,$GDDAT
055362  010037  001136                  MOV     R0,$BDADR
055366  062737  000024  001136          ADD     #RMMR1,$BDADR
055374  104275                          ERROR   275                     ;RUN AND GO NOT SET
055376  000534                          BR      420$
055400                          350$:
055400  004777  124124                  JSR     PC,@STOP                ;STOP THE CLOCK
                                ;STEP SEQUENCER TO LOCATION 156 (17 CLOCKS)
055404  012702  000021                  MOV     #17.,R2
055410                          360$:
```

```

055410 012760 141401 000024            MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

055416 012760 041401 000024            MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
055424 005302                          DEC     R2
055426 001370                          BNE     360$
                                ;DROP ON CYLINDER TO RESET LATCH, SET ON CYLINDER TO PASS TEST
                                ;AT LOCATION 166

055430 012760 041001 000024            MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
                                ;MOVE SEQUENCER TO SECTOR WAIT (34 CLOCKS)
055436 012702 000042                   MOV     #34.,R2
055442                          370$:

055442 012760 141401 000024            MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

055450 012760 041401 000024            MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
055456 005302                          DEC     R2
055460 001370                          BNE     370$
                                ;STEP THROUGH SECTOR WAIT LOOP TWICE AND VERIFY SEARCH IS ENABLED
                                ;DURING THE LOOP (6 CLOCKS)
055462 012702 000006                   MOV     #6.,R2
055466                          380$:

055466 012760 141401 000024            MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

055474 012760 041401 000024            MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1

055502 016037 000024 001142            MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
055510 042737 173777 001142            BIC     #↑CESRC,$BDDAT
055516 001403                          BEQ     390$
055520 005302                          DEC     R2
055522 001361                          BNE     380$
055524 000412                          BR      400$
055526 012737 004000 001140     390$:  MOV     #ESRC,$GDDAT
055534 010037 001136                   MOV     R0,$BDADR
055540 062737 000024 001136            ADD     #RMMR1,$BDADR
055546 104301                          ERROR   301             ;SEARCH NOT ENABLED DURING DATA
055550 000447                          BR      420$
055552                          400$:
                                ;SET DRIVE FAULT TO CAUSE ABORT CONDITION

055552 012760 041501 000024            MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
                                ;STEP SEQUENCER THROUGH ITS TEST FOR ABORT (2 CLOCKS)
055560 012702 000002                   MOV     #2,R2
055564                          410$:

055564 012760 141501 000024            MOV     #DMD!MUR!MOC!DBEN!MDF!DBCK,RMMR1(R0)    ;LOAD RMMR1

055572 012760 041501 000024            MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
055600 005302                          DEC     R2
055602 001370                          BNE     410$

                                ;ABORT EBL SHOULD NOW BE ACTIVE - USE THE MAINTENANCE REGISTER TO
                                ;FORCE BIT CLOCKS AND VERIFY THAT EBL SETS WITHIN 16 BIT CLOCKS
055604 012702 000020                   MOV     #16.,R2         ;MAXIMUM NUMBER OF BIT CLOCKS
```

```
055610                              415$:

055610  012760  045501  000024              MOV     #DMD!MUR!MOC!DBEN!MDF!MCLK,RMMR1(R0)    ;LOAD RMMR1

055616  012760  041501  000024              MOV     #DMD!MUR!MOC!DBEN!MDF,RMMR1(R0) ;LOAD RMMR1
                                    ;VERIFY EBL IS SET

055624  016037  000024  001142              MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
055632  042737  157777  001142              BIC     #↑CEBL,$BDDAT
055640  001013                              BNE     420$

055642  005302                              DEC     R2
055644  001361                              BNE     415$            ;CONTINUE BIT CLOCKS IF COUNT NOT 0
055646  012737  020000  001140              MOV     #EBL,$GDDAT
055654  010037  001136                      MOV     R0,$BDADR
055660  062737  000024  001136              ADD     #RMMR1,$BDADR
055666  104271                              ERROR   271             ;NO ABORT AT SECTOR WAIT

055670                              420$:                           ;END OF TEST


                                    ;;*****************************************************************
                                    ;*TEST 121          DATA COMMAND TESTS(3)

                                    ;;*****************************************************************
055670  000004                      TST121: SCOPE
055672  012737  000121  001226              MOV     #121,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX

055700  000240                              NOP
055702  012737  000024  001120              MOV     #20.,$ICNT      ;20 ITERATIONS
055710  112737  000001  001131              MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
055716  012737  055732  001122              MOV     #T121,$LPADR  ;LOAD LOOP ON TEST ADDRESS
055724  012737  055732  001124              MOV     #T121,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
055732                              T121:
055732  012706  001100                      MOV     #STACK,SP       ;LOAD THE STACK POINTER
055736  013700  001276                      MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
055742  013701  001456                      MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                    ;;*****************************************************************
                                    ;VERIFY THE TAG BUS DURING DATA COMMAND

                                    ;;*****************************************************************
                                    ;FIRST PART USES OFFSET FORWARD

                                    ;LOAD TEST PARAMETERS IN REGISTER OUTPUT BUFFER
055746  112737  000035  001410              MOVB    #29.,RMDA0      ;SECTOR 29
055754  112737  000004  001411              MOVB    #4,RMDA0+1      ;TRACK 4
055762  012737  001466  001436              MOV     #822.,RMDC0     ;CYLINDER 822
055770  012737  000200  001434              MOV     #OFD,RMOF0      ;FORWARD OFFSET
055776  012737  000071  001402              MOV     #RD!GO,RMCS10   ;READ DATA
056004  012737  177400  001404              MOV     #↑C256.+1,RMWC0 ;WORD COUNT
056012  012737  106652  001406              MOV     #BUFFER,RMBA0   ;BUFFER ADDRESS

                                    ;EXECUTE COMMAND AND VERIFY TAG BUS USING SUBROUTINE
056020  004737  056042                      JSR     PC,10$

                                    ;;*****************************************************************
                                    ;SECOND PART USES OFFSET REVERSE
```

```
                                           ;LOAD TEST PARAMETERS IN REGISTER OUTPUT BUFFER
056024  112737  000000  001434                     MOVB    #0,RMOFO         ;REVERSE OFFSET

                                           ;EXECUTE COMMAND AND VERIFY TAG BUS USING SUBROUTINE
056032  004737  056042                             JSR     PC,10$

056036  000137  057050                             JMP     300$

                                           ;;*************************************************************
                                           ;SUBROUTINE USED DURING TEST

056042                                     10$:
                                           ;SET VOLUME VALID USING SUBROUTINE
056042  004737  060526                             JSR     PC,SETVV                ;GO SET VOLUME VALID
056046  000403                                     BR      20$                     ;BRANCH TO 20$ IF NO ERROR
056050  104000                                     ERROR                           ;RETURN HERE IF ERROR
056052  000137  056666                             JMP     160$
056056                                     20$:
                                           ;LOAD TRACK, SECTOR AND CYLINDER ADDRESS, LOAD OFFSET

056056  013760  001410  000006                     MOV     RMDAO,RMDA(RO)   ;LOAD RMDA

056064  013760  001436  000034                     MOV     RMDCO,RMDC(RO)   ;LOAD RMDC

056072  013760  001434  000032                     MOV     RMOFO,RMOF(RO)   ;LOAD RMOF
                                           ;SET OFFSET MODE USING SUBROUTINE
056100  004737  060656                             JSR     PC,SETOM                ;GO SET OFFSET MODE
056104  000403                                     BR      30$                     ;BRANCH TO 30$ IF NO ERROR
056106  104000                                     ERROR                           ;RETURN HERE IF ERROR
056110  000137  056666                             JMP     160$
056114                                     30$:
                                           ;LOAD BUFFER ADDRESS AND WORD COUNT

056114  013760  001404  000002                     MOV     RMWCO,RMWC(RO)   ;LOAD RMWC

056122  013760  001406  000004                     MOV     RMBAO,RMBA(RO)   ;LOAD RMBA
                                           ;ENABLE DEBUG CLOCK AND LOAD DATA COMMAND

056130  012760  041401  000024                     MOV     #DMD!MUR!MOC!DBEN,RMMR1(RO)      ;LOAD RMMR1

056136  012760  000000  000014                     MOV     #0,RMER1(RO)     ;LOAD RMER1

056144  012760  000000  000042                     MOV     #0,RMER2(RO)     ;LOAD RMER2

056152  013760  001402  000000                     MOV     RMCS1O,RMCS1(RO)         ;LOAD RMCS1
                                           ;WAIT FOR RUN AND GO TO SET
056160  012737  000310  001524                     MOV     #200.,WATCH      ;SET WATCHDOG TIMER VALUE
056166  004777  123334                             JSR     PC,@CLOCK        ;START THE CLOCK
056172                                     40$:

056172  016037  000024  001142                     MOV     RMMR1(RO),$BDDAT         ;STORE RMMR1 AT $BDDAT
056200  042737  137777  001142                     BIC     #↑CRG,$BDDAT
056206  001020                                     BNE     50$
056210  005737  001524                             TST     WATCH
056214  001366                                     BNE     40$
056216  004777  123306                             JSR     PC,@STOP         ;STOP THE CLOCK
```

```
056222  012737  040000  001140          MOV     #RG,$GDDAT
056230  010037  001136                  MOV     R0,$BDADR
056234  062737  000024  001136          ADD     #RMMR1,$BDADR
056242  104275                          ERROR   275                 ;RUN AND GO NOT SET
056244  000137  056666                  JMP     160$
056250                          50$:
056250  004777  123254                  JSR     PC,@STOP            ;STOP THE CLOCK
056254  012704  056670                  MOV     #200$,R4            ;R4 = TABLE POINTER

                                ;STEP SEQUENCER TO HEAD SEQUENCE AT LOCATION 156 (17 CLOCKS)
056260  012705  000021                  MOV     #17.,R5             ;R5 = CLOCK COUNT
056264                          60$:

056264  016037  000040  001142          MOV     RMMR2(R0),$BDDAT            ;STORE RMMR2 AT $BDDAT
056272  042737  150000  001142          BIC     #RQA!RQB!TST,$BDDAT
056300  012437  001140                  MOV     (R4)+,$GDDAT
056304  053737  001436  001140          BIS     RMDCO,$GDDAT        ;OR CYLINDER ADDRESS
056312  023737  001140  001142          CMP     $GDDAT,$BDDAT
056320  001011                          BNE     70$                 ;BRANCH IF TAG BUS WRONG

056322  012760  141401  000024          MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)            ;LOAD RMMR1

056330  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)         ;LOAD RMMR1
056336  005305                          DEC     R5
056340  001351                          BNE     60$
056342  000407                          BR      80$
056344  010037  001136          70$:    MOV     R0,$BDADR
056350  062737  000040  001136          ADD     #RMMR2,$BDADR
056356  104276                          ERROR   276                 ;INCORRECT TAG BUS DURING DATA
056360  000542                          BR      160$
056362                          80$:
                                ;DROP ON CYLINDER TO RESET LATCH, SET ON CYLINDER TO PASS TEST AT
                                ;LOCATION 166

056362  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

056370  012760  041401  000024          MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)         ;LOAD RMMR1
                                ;STEP SEQUENCER TO END OF OFFSET AT LOCATION 174 (37 CLOCKS)
056376  012705  000045                  MOV     #37.,R5             ;RELOAD CLOCK COUNT
056402                          90$:

056402  016037  000040  001142          MOV     RMMR2(R0),$BDDAT            ;STORE RMMR2 AT $BDDAT
056410  042737  150000  001142          BIC     #RQA!RQB!TST,$BDDAT
056416  011437  001140                  MOV     (R4),$GDDAT
056422  032714  002000                  BIT     #CH,(R4)
056426  001430                          BEQ     110$                ;BRANCH IF CONTROL/HEADER NOT ON
056430  032714  004000                  BIT     #CC,(R4)
056434  001416                          BEQ     100$                ;BRANCH IF HEADER TAG
                                ;CONTROL TAG SHOULD BE ON-SETUP EXPECTED OFFSET
056436  052737  000010  001140          BIS     #BB03,$GDDAT        ;ASSUME OFD IS NOT SET
056444  032737  000200  001434          BIT     #OFD,RMOFO
056452  001406                          BEQ     95$
056454  042737  000010  001140          BIC     #BB03,$GDDAT        ;RESET BUS BIT 3 - DIRECTION IS REV
056462  052737  000004  001140          BIS     #BB02,$GDDAT
056470  000407                  95$:    BR      110$
```

```
056472                              100$:
                                    ;HEADER TAG SHOULD BE ON - SETUP EXPECTED TRACK ADDRESS
056472  013703  001410                      MOV     RMDA0,R3            ;GET TRACK
056476  000303                              SWAB    R3
056500  042703  177770                      BIC     #↑C7,R3
056504  050337  001140                      BIS     R3,$GDDAT
056510                              110$:
                                    ;COMPARE EXPECTED AND RECEIVED TAG BUS DATA
056510  023737  001140  001142              CMP     $GDDAT,$BDDAT
056516  001013                              BNE     120$

056520  012760  141401  000024              MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

056526  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
056534  062704  000002                      ADD     #2,R4               ;MOVE TABLE POINTER
056540  005305                              DEC     R5                  ;DECREMENT CLOCK COUNT
056542  001317                              BNE     90$
056544  000407                              BR      130$
056546  010037  001136              120$:   MOV     R0,$BDADR
056552  062737  000040  001136              ADD     #RMMR2,$BDADR
056560  104276                              ERROR   276                 ;INCORRECT TAG BUS DURING DATA
056562  000441                              BR      160$
056564                              130$:
                                    ;DROP ON CYLINDER TO RESET LATCH, RAISE ON CYLINDER TO PASS TEST AT
                                    ;SEQUENCER LOCATION 175

056564  012760  041001  000024              MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1

056572  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
                                    ;STEP SEQUENCER TO SECTOR WAIT LOOP (8 CLOCKS)
056600  012705  000010                      MOV     #8.,R5
056604                              140$:

056604  016037  000040  001142              MOV     RMMR2(R0),$BDDAT            ;STORE RMMR2 AT $BDDAT
056612  042737  150000  001142              BIC     #RQA!RQB!TST,$BDDAT
056620  023737  001140  001142              CMP     $GDDAT,$BDDAT       ;GOOD DATA SAME AS LAST CMP
056626  001011                              BNE     150$                ;BRANCH IF ERROR

056630  012760  141401  000024              MOV     #DMD!MUR!MOC!DBEN!DBCK,RMMR1(R0)        ;LOAD RMMR1

056636  012760  041401  000024              MOV     #DMD!MUR!MOC!DBEN,RMMR1(R0)     ;LOAD RMMR1
056644  005305                              DEC     R5
056646  001356                              BNE     140$
056650  000406                              BR      160$
056652  010037  001136              150$:   MOV     R0,$BDADR
056656  062737  000040  001136              ADD     #RMMR2,$BDADR
056664  104276                              ERROR   276                 ;INCORRECT TAG BUS DURING DATA

056666  000207                      160$:   RTS     PC

056670                              200$:
                                    ;TABLE OF TAG BUS CONTROL AND DATA VALUES
056670  001777                              .WORD   1777                ;LOCATION 0
056672  001777                              .WORD   1777                ;LOCATION 25
056674  001777                              .WORD   1777                ;LOCATION 26
056676  001777                              .WORD   1777                ;LOCATION 128
```

```
056700  001777          .WORD   1777            ;LOCATION 129
056702  001777          .WORD   1777            ;LOCATION 130
056704  004000          .WORD   CC              ;LOCATION 144
056706  004000          .WORD   CC              ;LOCATION 145
056710  024000          .WORD   CC!TAG          ;LOCATION 146
056712  024000          .WORD   CC!TAG          ;LOCATION 147
056714  024000          .WORD   CC!TAG          ;LOCATION 148
056716  024000          .WORD   CC!TAG          ;LOCATION 149
056720  024000          .WORD   CC!TAG          ;LOCATION 150
056722  024000          .WORD   CC!TAG          ;LOCATION 151
056724  004000          .WORD   CC              ;LOCATION 152
056726  004000          .WORD   CC              ;LOCATION 153
056730  001777          .WORD   1777            ;LOCATION 154
056732  002000          .WORD   CH              ;LOCATION 156
056734  002000          .WORD   CH              ;LOCATION 157
056736  022000          .WORD   CH!TAG          ;LOCATION 158
056740  022000          .WORD   CH!TAG          ;LOCATION 159
056742  022000          .WORD   CH!TAG          ;LOCATION 160
056744  022000          .WORD   CH!TAG          ;LOCATION 161
056746  022000          .WORD   CH!TAG          ;LOCATION 162
056750  022000          .WORD   CH!TAG          ;LOCATION 163
056752  002000          .WORD   CH              ;LOCATION 164
056754  002000          .WORD   CH              ;LOCATION 165
056756  001777          .WORD   1777            ;LOCATION 232
056760  001777          .WORD   1777            ;LOCATION 233
056762  001777          .WORD   1777            ;LOCATION 234
056764  001777          .WORD   1777            ;LOCATION 235
056766  001777          .WORD   1777            ;LOCATION 236
056770  001777          .WORD   1777            ;LOCATION 237
056772  001777          .WORD   1777            ;LOCATION 238
056774  001777          .WORD   1777            ;LOCATION 239
056776  001777          .WORD   1777            ;LOCATION 240
057000  001777          .WORD   1777            ;LOCATION 241
057002  001777          .WORD   1777            ;LOCATION 242
057004  001777          .WORD   1777            ;LOCATION 243
057006  001777          .WORD   1777            ;LOCATION 244
057010  001777          .WORD   1777            ;LOCATION 245
057012  001777          .WORD   1777            ;LOCATION 246
057014  001777          .WORD   1777            ;LOCATION 247
057016  001777          .WORD   1777            ;LOCATION 248
057020  001777          .WORD   1777            ;LOCATION 249
057022  001777          .WORD   1777            ;LOCATION 250
057024  001777          .WORD   1777            ;LOCATION 251
057026  001777          .WORD   1777            ;LOCATION 252
057030  001777          .WORD   1777            ;LOCATION 166
057032  006000          .WORD   CC!CH           ;LOCATION 169
057034  006000          .WORD   CC!CH           ;LOCATION 170
057036  026000          .WORD   CC!CH!TAG       ;LOCATION 171
057040  026000          .WORD   CC!CH!TAG       ;LOCATION 172
057042  026000          .WORD   CC!CH!TAG       ;LOCATION 173
057044  026000          .WORD   CC!CH!TAG       ;LOCATION 174
057046  026000          .WORD   CC!CH!TAG       ;LOCATION 175,ETC

057050          300$:                           ;END OF TEST
```

```
057050                                  $EOSP:
057050  000240                                  NOP
057052  013700  001456                          MOV     TSTQUE,R0
057056  062700  000002                          ADD     #2,R0
057062  010037  001456                          MOV     R0,TSTQUE
057066  005710                                  TST     (R0)
057070  001402                                  BEQ     1$
057072  000137  006212                          JMP     READY
057076  012737  001460  001456  1$:     MOV     #TSTQUE+2,TSTQUE
                                        .SBTTL  END OF PASS ROUTINE

                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
                                        ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO READY

057104                                  $EOP:
057104  000240                                  NOP
057106  005037  001116                          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
057112  005037  001206                          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
057116  005237  001230                          INC     $PASS           ;;INCREMENT THE PASS NUMBER
057122  042737  100000  001230                  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
057130  005327                                  DEC     (PC)+           ;;LOOP?
057132  000001                          $EOPCT: .WORD   1
057134  003063                                  BGT     $DOAGN          ;;YES
057136  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
057140  000001                          $ENDCT: .WORD   1
057142  057132                                  $EOPCT
057144  104401  057152                          TYPE    ,65$            ;;TYPE ASCIZ STRING
057150  000407                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <12><15>/END PASS #/
057170                                  64$:
057170  013746  001230                          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                        ;;TYPE PASS NUMBER
057174  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
057176  104401  057204                          TYPE    ,67$            ;;TYPE ASCIZ STRING
057202  000421                                  BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
057246                                  66$:
057246  013746  001126                          MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                        ;;TOTAL NUMBER OF ERRORS
057252  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
057254  104401  001217                          TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
057260  005037  001126                          CLR     $ERTTL          ;;CLEAR ERROR TOTAL
057264  013700  000042                  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
057270  001405                                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
057272  000005                                  RESET                   ;;CLEAR THE WORLD
057274  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
057276  000240                                  NOP                     ;;SAVE ROOM
057300  000240                                  NOP                     ;;FOR
057302  000240                                  NOP                     ;;ACT11
057304                                  $DOAGN:
057304  000137                                  JMP     @(PC)+          ;;RETURN
057306  006212                          $RTNAD: .WORD   READY
```

```
 13694  057310     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 13695             057314                        .EVEN
```

```
                                        .SBTTL  SUBROUTINES

                                        ;;*****************************************************************
                                        .SBTTL  ERROR TYPEOUT ROUTINE

                                        ;*THE ERROR TYPEOUT ROUTINE ASSEMBLES AND PRINTS INFORMATION
                                        ;*REGARDING THE DETECTION OF AN ERROR AS FOLLOWS:
                                        ;*
                                        ;*       .UNIT NUMBER, TEST NUMBER, ERROR NUMBER AND PROGRAM COUNTER ARE
                                        ;*PRINTED ON THE FIRST LINE;
                                        ;*       .ERROR MESSAGE IS ASSEMBLED, FORMATTED AND PRINTED ON
                                        ;*ONE OR MORE SUCCEEDING LINES;
                                        ;*       .PAIRED LINES OF ERROR HEADERS AND ERROR DATA
                                        ;*ARE PRINTED AFTER THE ERROR MESSAGE.

057314                                  ERRTYP:
057314  104414                                  SAVREG
057316  032777  020000  121630                  BIT     #SW13,@SWR              ;INHIBIT TYPEOUTS??
057324  001402                                  BEQ     1$                      ;NO!!
057326  000137  060044                          JMP     21$                     ;YES!!
                                        ;TYPE UNIT NUMBER, TEST NUMBER, ERROR NUMBER, AND PROGRAM COUNTER
057332  104401  001217                  1$:     TYPE    ,$CRLF
057336  104401  060060                          TYPE    ,ERTY00                 ;TYPE "UNT#"
057342  013746  001234                          MOV     $UNIT,-(SP)             ;;SAVE $UNIT FOR TYPEOUT
                                                                                ;;TYPE UNIT NUMBER
057346  104403                                  TYPOS                           ;;GO TYPE--OCTAL ASCII
057350     003                                  .BYTE   3                       ;;TYPE 3 DIGIT(S)
057351     000                                  .BYTE   0                       ;;SUPPRESS LEADING ZEROS
057352  005037  060050                          CLR     TSTNMB                  ;LOAD TEST NUMBER FOR
057356  013737  001226  060050                  MOV     $TESTN,TSTNMB
057364  104401  060066                          TYPE    ,ERTY01                 ;TYPE "TST#"
057370  013746  060050                          MOV     TSTNMB,-(SP)            ;;SAVE TSTNMB FOR TYPEOUT
                                                                                ;;TYPE TEST NUMBER
057374  104403                                  TYPOS                           ;;GO TYPE--OCTAL ASCII
057376     003                                  .BYTE   3                       ;;TYPE 3 DIGIT(S)
057377     000                                  .BYTE   0                       ;;SUPPRESS LEADING ZEROS
057400  005037  060052                          CLR     ERRNMB                  ;LOAD ERROR NUMBER FOR
057404  113737  001130  060052                  MOVB    $ITEMB,ERRNMB           ;TYPEOUT
057412  001406                                  BEQ     2$                      ;SKIP IF NO ERROR CALLED
057414  104401  060076                          TYPE    ,ERTY02                 ;TYPE "ERR#"
057420  013746  060052                          MOV     ERRNMB,-(SP)            ;;SAVE ERRNMB FOR TYPEOUT
                                                                                ;;TYPE ERROR NUMBER
057424  104403                                  TYPOS                           ;;GO TYPE--OCTAL ASCII
057426     003                                  .BYTE   3                       ;;TYPE 3 DIGIT(S)
057427     000                                  .BYTE   0                       ;;SUPPRESS LEADING ZEROS
057430  104401  060105                  2$:     TYPE    ,ERTY03                 ;TYPE "PC="
057434  013746  001132                          MOV     $ERRPC,-(SP)            ;;SAVE $ERRPC FOR TYPEOUT
                                                                                ;;TYPE PROGRAM COUNTER
057440  104403                                  TYPOS                           ;;GO TYPE--OCTAL ASCII
057442     006                                  .BYTE   6                       ;;TYPE 6 DIGIT(S)
057443     001                                  .BYTE   1                       ;;TYPE LEADING ZEROS
                                        ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
057444  005737  060052                  3$:     TST     ERRNMB                  ;WAS AN ERROR CALLED?
057450  001575                                  BEQ     21$                     ;NO!!
057452  104401  001217                          TYPE    ,$CRLF                  ;YES-TYPE CRLF
057456  105037  060056                          CLRB    BOTFLG                  ;CLEAR BOT FLAG
```

```
057462  105037  060057                   CLRB    CHRCNT            ;CLEAR CHARACTER COUNTER
057466  013700  060052                   MOV     ERRNMB,R0         ;R0 POINTS TO FIRST OF
057472  006300                           ASL     R0                ;FOUR ENTRIES IN ERROR
057474  006300                           ASL     R0                ;TABLE
057476  006300                           ASL     R0
057500  062700  001522                   ADD     #$ERRTB-8.,R0
057504  011001                           MOV     (R0),R1           ;R1 POINTS TO ERROR MESSAGE
                                                                   ;TABLE
057506  001507                           BEQ     12$               ;BRANCH IF NO ERROR MESSAGE
                                 ;TYPE THE ERROR MESSAGE
057510  012102                   4$:     MOV     (R1)+,R2          ;R2=ADDRESS OF MESSAGE STRING
057512  001505                           BEQ     12$               ;BRANCH IF END OF MESSAGE
057514  010237  057662                   MOV     R2,11$            ;LOAD ADDRESS OF STRING
057520  005037  060054                   CLR     BOTADR            ;CLEAR BOT ADDRESS
057524  112203                   5$:     MOVB    (R2)+,R3          ;END OF STRING??
057526  001454                           BEQ     10$               ;YES!!
057530  122703  000015                   CMPB    #CR,R3            ;CARRIAGE RETURN??
057534  001003                           BNE     6$                ;NO!!
057536  105037  060057                   CLRB    CHRCNT            ;YES-CLEAR CHAR COUNT
057542  000770                           BR      5$                ;GET NEXT CHARACTER
057544  122703  000012           6$:     CMPB    #LF,R3            ;LINE FEED??
057550  001765                           BEQ     5$                ;YES-GET NEXT CHARACTER
057552  122703  000011                   CMPB    #HT,R3            ;HORIZONTAL TAB??
057556  001007                           BNE     8$                ;NO!!
057560  105237  060057           7$:     INCB    CHRCNT            ;ADJUST CHARACTER COUNT
057564  132737  000007  060057           BITB    #7,CHRCNT
057572  001372                           BNE     7$
057574  000407                           BR      9$
057576  105237  060057           8$:     INCB    CHRCNT            ;INCREMENT CHARACTER COUNT
057602  122703  000040                   CMPB    #' ,R3            ;SPACE??
057606  001002                           BNE     9$                ;NO!!
057610  010237  060054                   MOV     R2,BOTADR         ;SAVE ADDRESS OF SPACE
057614  122737  000100  060057   9$:     CMPB    #64.,CHRCNT       ;END OF LINE??
057622  103340                           BHIS    5$                ;NO!!
057624  013704  060054                   MOV     BOTADR,R4         ;GET ADDRESS OF LAST SPACE
057630  001007                           BNE     90$               ;BRANCH IF SPACE DETECTED
057632  104401  001217                   TYPE    ,$CRLF            ;TYPE CRLF
057636  105037  060057                   CLRB    CHRCNT            ;CLEAR CHARACTER COUNT
057642  013702  057662                   MOV     11$,R2            ;SET UP R2 FOR TESTING
057646  000726                           BR      5$
057650  105044                   90$:    CLRB    -(R4)             ;REPLACE SPACE
057652  112737  177777  060056           MOVB    #-1,BOTFLG        ;SET BOT FLAG
057660  104401                   10$:    TYPE                      ;TYPE ERROR MESSAGE STRING
057662  000000                   11$:    .WORD                     ;STRING ADDRESS GOES HERE
057664  105737  060056                   TSTB    BOTFLG            ;WAS STRING TRUNCATED??
057670  001707                           BEQ     4$                ;NO!!
057672  104401  001217                   TYPE    ,$CRLF            ;YES-TYPE CRLF
057676  105037  060056                   CLRB    BOTFLG            ;CLEAR BOT FLAG
057702  105037  060057                   CLRB    CHRCNT            ;CLEAR CHARACTER COUNT
057706  013702  060054                   MOV     BOTADR,R2         ;SETUP R2 FOR TESTING
057712  010237  057662                   MOV     R2,11$            ;SETUP 11$ FOR TYPING
057716  112742  000040                   MOVB    #' ,-(R2)         ;RESTORE SPACE
057722  105722                           TSTB    (R2)+             ;RESTORE R2
057724  000677                           BR      5$                ;TYPE REST OF STRING
057726                           12$:
                                 ;TYPE ERROR HEADER AND ERROR DATA
```

```
057726                          13$:
057726  016001  000002                  MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
057732  001444                          BEQ     21$             ;BRANCH IF NO HEADER
057734  104401  001217                  TYPE    ,$CRLF          ;(ASSUME NO DATA)
057740  016002  000004                  MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
057744  016003  000006                  MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
057750  012137  057760          14$:    MOV     (R1)+,15$       ;PUT HEADER ADDRESS FOR TYPE
057754  001433                          BEQ     21$             ;BRANCH IF END OF HEADERS
                                                                ;(ASSUME END OF DATA)
057756  104401                          TYPE
057760  000000                  15$:    .WORD   0               ;HEADER ADDRESS GOES HERE
057762  104401  001217                  TYPE    ,$CRLF
057766  005702                          TST     R2              ;DATA WITH HEADER??
057770  001767                          BEQ     14$             ;NO!!
057772  012204                          MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
057774  012305                          MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
057776  105725                  16$:    TSTB    (R5)+           ;WHAT KIND OF DATA??
060000  100407                          BMI     18$             ;BINARY
060002  001403                          BEQ     17$             ;OCTAL
060004  013446                          MOV     @(R4)+,-(SP)    ;DECIMAL
060006  104405                          TYPDS
060010  000405                          BR      19$
060012  013446                  17$:    MOV     @(R4)+,-(SP)
060014  104402                          TYPOC
060016  000402                          BR      19$
060020  013446                  18$:    MOV     @(R4)+,-(SP)
060022  104406                          TYPBN
060024  005714                  19$:    TST     (R4)            ;MORE DATA??
060026  001403                          BEQ     20$             ;NO!!
060030  104401  060113                  TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
060034  000760                          BR      16$             ;AND CONTINUE
060036  104401  001217          20$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
060042  000742                          BR      14$             ;BEFORE NEXT HEADER
060044  104415                  21$:    RESREG
060046  000207                          RTS     PC

060050  000000                  TSTNMB: .WORD   0               ;TEST NUMBER
060052  000000                  ERRNMB: .WORD   0               ;ERROR NUMBER
060054  000000                  BOTADR: .WORD   0               ;BEGINNING OF TEXT ADDRESS
060056     000                  BOTFLG: .BYTE   0               ;BOT FLAG
060057     000                  CHRCNT: .BYTE   0               ;CHARACTER COUNT

060060  047125  052111  000043  ERTY00: .ASCIZ  @UNIT#@
060066  020054  042524  052123  ERTY01: .ASCIZ  @, TEST#@
060074  000043
060076  020054  051105  021522  ERTY02: .ASCIZ  @, ERR#@
060104     000
060105     054  050040  036503  ERTY03: .ASCIZ  @, PC=@
060112     000
060113     040  000040          ERTY04: .ASCIZ  @  @
                                        .EVEN
```

```
                                        .SBTTL  CLOCK SUBROUTINES

                                        ;ROUTINE TO SIZE FOR CLOCKS (KW11-L OR KW11-P)
                                        ;;*************************************************************
060116  000240                          SIZCLK: NOP
060120  013746  000004                          MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
060124  013746  000006                          MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
060130  012737  060214  000004                  MOV     #10$,ERRVEC     ;LOAD 04 TRAP VECTORS
060136  012737  000300  000006                  MOV     #PR6,ERRVEC+2

                                        ;SEE IF A KW11-P CLOCK IS PRESENT - GO TO 10$ IF NOT PRESENT
060144  005777  121332                          TST     @$LPCSR         ;TEST FOR P CLOCK
060150  012737  060356  001526                  MOV     #PCLOCK,CLOCK   ;LOAD SUBROUTINE ADDRESS
060156  012737  060500  001530                  MOV     #PSTOP,STOP     ;LOAD STOP ADDRESS
060164  012777  060444  121314                  MOV     #PCOUNT,@$LPVEC ;LOAD P CLOCK INTERRUPT VECTOR
060172  012777  000300  121310                  MOV     #PR6,@$LPVEC+2
060200  013777  001516  121306                  MOV     $LLVEC+2,@$LLVEC;CLEAR L CLOCK INTERRUPT VECTOR
060206  005077  121304                          CLR     @$LLVEC+2
060212  000454                                  BR      30$
060214  012716  060222                  10$:    MOV     #15$,(SP)       ;DUMMY RTI ADDRESS
060220  000002                                  RTI                     ;RESTORE PRIORITY
060222                                  15$:
                                        ;NO P CLOCK-SEE IF L CLOCK IS PRESENT-GO TO 20$ IF NOT PRESET
060222  012737  060300  000004                  MOV     #20$,ERRVEC     ;CHANGE 04 TRAP VECTOR
060230  005777  121256                          TST     @$LLCSR         ;TEST FOR L CLOCK
060234  012737  060374  001526                  MOV     #LCLOCK,CLOCK   ;LOAD SUBROUTINE ADDRESS
060242  012737  060506  001530                  MOV     #LSTOP,STOP     ;LOAD STOP ADDRESS
060250  012777  060444  121236                  MOV     #LCOUNT,@$LLVEC ;LOAD L CLOCK INTERRUPT VECTOR
060256  012777  000300  121232                  MOV     #PR6,@$LLVEC+2
060264  013777  001510  121214                  MOV     $LPVEC+2,@$LPVEC;CLEAR P CLOCK INTERRUPT VECTOR
060272  005077  121212                          CLR     @$LPVEC+2
060276  000422                                  BR      30$
060300  012716  060306                  20$:    MOV     #25$,(SP)       ;DUMMY RTI ADDRESS
060304  000002                                  RTI                     ;RESTORE PRIORITY
060306                                  25$:
                                        ;NO CLOCK AVAILABLE - AUGMENT RETURN ADDRESS
060306  005037  001526                          CLR     CLOCK           ;CLEAR SUBROUTINE ADDRESS
060312  012737  001510  001506                  MOV     #$LPVEC+2,$LPVEC;CLEAR P CLOCK INTERRUPT VECTOR
060320  005037  001510                          CLR     $LPVEC+2
060324  012737  001516  001514                  MOV     #$LLVEC+2,$LLVEC;CLEAR L CLOCK INTERRUPT VECTOR
060332  005037  001516                          CLR     $LLVEC+2
060336  062766  000002  000004                  ADD     #2,4(SP)        ;CHANGE RETURN ADDRESS

060344                                  30$:
060344  012637  000006                          MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
060350  012637  000004                          MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
060354  000207                                  RTS     PC
                                        ;ROUTINES TO START THE CLOCK (KW11-L OR KW11-P)
                                        ;;*************************************************************

060356  012777  177777  121120          PCLOCK: MOV     #-1,@$LPCSB     ;LOAD COUNT SET BUFFER
060364  012777  000135  121110                  MOV     #135,@$LPCSR    ;LOAD CONTROL REGISTER
060372  000403                                  BR      PLCLK           ;GO TO COMMON CODE

060374  012777  000100  121110          LCLOCK: MOV     #100,@$LLCSR    ;LOAD CONTROL REGISTER
```

```
060402  005037  001522          PLCLK:  CLR     TIME            ;CLEAR TIMER COUNT
060406  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
060410  012605                          MOV     (SP)+,R5        ;;SAVE THE PSW IN R5
060412  010537  001520                  MOV     R5,$PSW         ;SAVE PRIORITY
060416  042705  177437                  BIC     #↑CPR7,R5       ;MASK X
060422  022705  000300                  CMP     #PR6,R5         ;IS PRIORITY TOO HIGH??
060426  101005                          BHI     40$             ;NO!!
060430  012746  000240                  MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
060434  012746  060442                  MOV     #30$,-(SP)      ;;PUT NEW PC ON  STACK
060440  000002                          RTI                     ;;POP NEW PC AND PS
060442                          30$:
060442  000207                  40$:    RTS     PC

                                ;ROUTINES TO HANDLE CLOCK INTERRUPTS (KW11-L OR KW11-P)

060444                          PCOUNT:
060444                          LCOUNT:
060444  062737  000021  001522          ADD     #17.,TIME       ;ADD 17MS TO ELAPSED TIME
060452  103003                          BCC     10$             ;BRANCH IF NO OVERFLOW
060454  012737  177777  001522          MOV     #-1,TIME        ;RESTORE MAXIMUM COUNT
060462  162737  000021  001524  10$:    SUB     #17.,WATCH      ;DECREMENT REMAINING TIME
060470  100002                          BPL     20$             ;BRANCH IF POSITIVE
060472  005037  001524                  CLR     WATCH           ;CLEAR REMAINING TIME
060476  000002                  20$:    RTI                     ;RETURN TO USER

                                ;ROUTINES TO STOP THE CLOCK (KW11-L OR KW11-P)

060500  005077  120776          PSTOP:  CLR     @$LPCSR         ;STOP P CLOCK
060504  000402                          BR      PLSTP           ;GO TO COMMON STOP CODE

060506  005077  121000          LSTOP:  CLR     @$LLCSR         ;STOP L CLOCK

060512                          PLSTP:
060512  013746  001520                  MOV     $PSW,-(SP)      ;;PUT NEW PS ON STACK
060516  012746  060524                  MOV     #10$,-(SP)      ;;PUT NEW PC ON  STACK
060522  000002                          RTI                     ;;POP NEW PC AND PS
060524                          10$:
060524  000207                          RTS     PC

                                .SBTTL  SET VOLUME VALID SUBROUTINE

                                ;THIS SUBROUTINE INITIALIZES THE SUBSYSTEM AND SETS VOLUME VALID,
                                ;RETURNING WITH THE DRIVE STILL IN DIAGNOSTIC MODE.  THE SUBROUTINE
                                ;RETURNS TO THE WORD FOLLOWING THE CALL, EXCEPT WHEN AN ERROR IS
                                ;DETECTED, IN WHICH CASE IT RETURNS TO THE SECOND WORD FOLLOWING THE
                                ;CALL.
                                ;
                                ;CALL:  JSR     PC,SETVV        JUMP TO SUBROUTINE
                                ;       BR      ??              RETURN HERE IF NO ERROR
                                ;       ERROR                   RETURN HERE IF ERROR
                                ;
060526                          SETVV:
060526  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
060534  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
060540  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
```

```
060546  012760  001001  000024          MOV     #DMD!MUR,RMMR1(R0)          ;LOAD RMMR1

060554  012760  000000  000014          MOV     #0,RMER1(R0)        ;LOAD RMER1

060562  012760  000000  000042          MOV     #0,RMER2(R0)        ;LOAD RMER2

060570  012760  000023  000000          MOV     #PACACK!GO,RMCS1(R0)        ;LOAD RMCS1

060576  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
060604  042737  177677  001142          BIC     #^CVV,$BDDAT
060612  001020                          BNE     10$             ;BRANCH IF VOLUME VALID SET
060614  010037  001136                  MOV     R0,$BDADR       ;SETUP FOR ERROR MSG
060620  062737  000012  001136          ADD     #RMDS,$BDADR
060626  012737  000100  001140          MOV     #VV,$GDDAT
060634  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN ADDRESS TO ERROR
060640  112776  000170  000000          MOVB    #170,@(SP)      ;WRITE ERROR NUMBER
060646  012737  000022  001174          MOV     #PACACK,$TMP0
060654  000207                  10$:    RTS     PC              ;RETURN

                                .SBTTL  SET OFFSET MODE SUBROUTINE

                                ;THIS SUBROUTINE EXECUTES AN OFFSET COMMAND AND VERIFIES THAT OFFSET
                                ;MODE SETS.  THE DRIVE SHOULD BE IN DIAGNOSTIC MODE WHEN CALLING THE
                                ;SUBROUTINE, WHICH WILL LEAVE DMD ON.  THE SUBROUTINE RETURNS TO THE
                                ;WORD FOLLOWING THE CALL UNLESS THERE IS AN ERROR, IN WHICH CASE IT
                                ;RETURNS TO THE SECOND WORD FOLLOWING THE CALL
                                ;
                                ;CALL:  JSR     PC,SETOM        JUMP TO SUBROUTINE
                                ;       BR      ??              RETURN HERE IF NO ERROR
                                ;       ERROR                   RETURN HERE IF ERROR

060656                          SETOM:

060656  012760  001001  000024          MOV     #DMD!MUR,RMMR1(R0)          ;LOAD RMMR1

060664  012760  000000  000014          MOV     #0,RMER1(R0)        ;LOAD RMER1

060672  012760  000000  000042          MOV     #0,RMER2(R0)        ;LOAD RMER2

060700  012760  000015  000000          MOV     #OFFSET!GO,RMCS1(R0)        ;LOAD RMCS1

060706  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
060714  042737  177776  001142          BIC     #^COM,$BDDAT
060722  001015                          BNE     10$             ;BRANCH IF OFFSET ON
060724  012737  000001  001140          MOV     #OM,$GDDAT
060732  010037  001136                  MOV     R0,$BDADR
060736  062737  000012  001136          ADD     #RMDS,$BDADR
060744  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN ADDRESS TO ERROR
060750  112776  000200  000000          MOVB    #200,@(SP)      ;WRITE ERROR NUMBER
060756                          10$:
060756  000207                          RTS     PC              ;RETURN TO USER
```

```
                                .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

 060760                         $SAVREG:
 060760  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 060762  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 060764  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 060766  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 060770  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 060772  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 060774  016646  000022                 MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 061000  016646  000022                 MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 061004  016646  000022                 MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 061010  016646  000022                 MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 061014  000002                         RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
 061016                         $RESREG:
 061016  012666  000022                 MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 061022  012666  000022                 MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 061026  012666  000022                 MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 061032  012666  000022                 MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 061036  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
 061040  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
 061042  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
 061044  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
 061046  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 061050  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 061052  000002                         RTI
                                .SBTTL  BINARY TO ASCII AND TYPE ROUTINE

                                ;;**********************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
                                ;*BINARY-ASCII NUMBER AND TYPE IT.
                                ;*CALL:
                                ;*      MOV     NUMBER,-(SP)    ;;NUMBER TO BE TYPED
                                ;*      TYPBN                   ;;TYPE IT

 061054  010146                 $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
 061056  016601  000006                 MOV     6(SP),R1        ;;GET THE INPUT NUMBER
```

```
061062  000261                          SEC                             ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
061064  112737  000060  061126  1$:     MOVB    #'0,$BIN                ;;SET CHARACTER TO AN ASCII "0".
061072  006101                          ROL     R1                      ;;GET THIS BIT
061074  001406                          BEQ     2$                      ;;DONE?
061076  105537  061126                  ADCB    $BIN                    ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
061102  104401  061126                  TYPE    ,$BIN                   ;;GO TYPE THIS BIT
061106  000241                          CLC                             ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
061110  000765                          BR      1$                      ;;GO DO THE NEXT BIT
061112  012601                  2$:     MOV     (SP)+,R1                ;;POP THE STACK INTO R1
061114  016666  000002  000004          MOV     2(SP),4(SP)             ;;ADJUST THE STACK
061122  012616                          MOV     (SP)+,(SP)
061124  000002                          RTI                             ;;RETURN TO USER
061126     000     000          $BIN:   .BYTE   0,0                     ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;*******************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)               ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                           ;;GO TO THE ROUTINE

061130                          $TYPDS:
061130  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
061132  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
061134  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
061136  010346                          MOV     R3,-(SP)                ;;PUSH R3 ON STACK
061140  010546                          MOV     R5,-(SP)                ;;PUSH R5 ON STACK
061142  012746  020200                  MOV     #20200,-(SP)            ;;SET BLANK SWITCH AND SIGN
061146  016605  000020                  MOV     20(SP),R5               ;;GET THE INPUT NUMBER
061152  100004                          BPL     1$                      ;;BR IF INPUT IS POS.
061154  005405                          NEG     R5                      ;;MAKE THE BINARY NUMBER POS.
061156  112766  000055  000001          MOVB    #'-,1(SP)               ;;MAKE THE ASCII NUMBER NEG.
061164  005000                  1$:     CLR     R0                      ;;ZERO THE CONSTANTS INDEX
061166  012703  061344                  MOV     #$DBLK,R3               ;;SETUP THE OUTPUT POINTER
061172  112723  000040                  MOVB    #' ,(R3)+               ;;SET THE FIRST CHARACTER TO A BLANK
061176  005002                  2$:     CLR     R2                      ;;CLEAR THE BCD NUMBER
061200  016001  061334                  MOV     $DTBL(R0),R1            ;;GET THE CONSTANT
061204  160105                  3$:     SUB     R1,R5                   ;;FORM THIS BCD DIGIT
061206  002402                          BLT     4$                      ;;BR IF DONE
061210  005202                          INC     R2                      ;;INCREASE THE BCD DIGIT BY 1
061212  000774                          BR      3$
061214  060105                  4$:     ADD     R1,R5                   ;;ADD BACK THE CONSTANT
061216  005702                          TST     R2                      ;;CHECK IF BCD DIGIT=0
061220  001002                          BNE     5$                      ;;FALL THROUGH IF 0
061222  105716                          TSTB    (SP)                    ;;STILL DOING LEADING 0'S?
061224  100407                          BMI     7$                      ;;BR IF YES
061226  106316                  5$:     ASLB    (SP)                    ;;MSD?
061230  103003                          BCC     6$                      ;;BR IF NO
061232  116663  000001  177777          MOVB    1(SP),-1(R3)            ;;YES--SET THE SIGN
061240  052702  000060          6$:     BIS     #'0,R2                  ;;MAKE THE BCD DIGIT ASCII
061244  052702  000040          7$:     BIS     #' ,R2                  ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
061250  110223                          MOVB    R2,(R3)+                ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
```

```
061252  005720                          TST     (R0)+                   ;;JUST INCREMENTING
061254  020027  000010                  CMP     R0,#10                  ;;CHECK THE TABLE INDEX
061260  002746                          BLT     2$                      ;;GO DO THE NEXT DIGIT
061262  003002                          BGT     8$                      ;;GO TO EXIT
061264  010502                          MOV     R5,R2                   ;;GET THE LSD
061266  000764                          BR      6$                      ;;GO CHANGE TO ASCII
061270  105726                  8$:     TSTB    (SP)+                   ;;WAS THE LSD THE FIRST NON-ZERO?
061272  100003                          BPL     9$                      ;;BR IF NO
061274  116663  177777  177776          MOVB    -1(SP),-2(R3)           ;;YES--SET THE SIGN FOR TYPING
061302  105013                  9$:     CLRB    (R3)                    ;;SET THE TERMINATOR
061304  012605                          MOV     (SP)+,R5                ;;POP STACK INTO R5
061306  012603                          MOV     (SP)+,R3                ;;POP STACK INTO R3
061310  012602                          MOV     (SP)+,R2                ;;POP STACK INTO R2
061312  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
061314  012600                          MOV     (SP)+,R0                ;;POP STACK INTO R0
061316  104401  061344                  TYPE    ,$DBLK                  ;;NOW TYPE THE NUMBER
061322  016666  000002  000004          MOV     2(SP),4(SP)             ;;ADJUST THE STACK
061330  012616                          MOV     (SP)+,(SP)
061332  000002                          RTI                             ;;RETURN TO USER
061334  023420                  $DTBL:  10000.
061336  001750                          1000.
061340  000144                          100.
061342  000012                          10.
061344  000004                  $DBLK:  .BLKW   4
                                .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

061354  017646  000000          $TYPOS: MOV     @(SP),-(SP)             ;;PICKUP THE MODE
061360  116637  000001  061577          MOVB    1(SP),$0FILL            ;;LOAD ZERO FILL SWITCH
061366  112637  061601                  MOVB    (SP)+,$OMODE+1          ;;NUMBER OF DIGITS TO TYPE
061372  062716  000002                  ADD     #2,(SP)                 ;;ADJUST RETURN ADDRESS
061376  000406                          BR      $TYPON
061400  112737  000001  061577  $TYPOC: MOVB    #1,$0FILL               ;;SET THE ZERO FILL SWITCH
061406  112737  000006  061601          MOVB    #6,$OMODE+1             ;;SET FOR SIX(6) DIGITS
```

```
061414  112737  000005  061576  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
061422  010346                          MOV     R3,-(SP)        ;;SAVE R3
061424  010446                          MOV     R4,-(SP)        ;;SAVE R4
061426  010546                          MOV     R5,-(SP)        ;;SAVE R5
061430  113704  061601                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
061434  005404                          NEG     R4
061436  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
061442  110437  061600                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
061446  113704  061577                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
061452  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
061456  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
061460  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
061462  000404                          BR      3$              ;;GO DO MSB
061464  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
061466  006105                          ROL     R5
061470  006105                          ROL     R5
061472  010503                          MOV     R5,R3
061474  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
061476  105337  061600                  DECB    $OMODE          ;;TYPE THIS DIGIT?
061502  100016                          BPL     7$              ;;BR IF NO
061504  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
061510  001002                          BNE     4$              ;;TEST FOR 0
061512  005704                          TST     R4              ;;SUPPRESS THIS 0?
061514  001403                          BEQ     5$              ;;BR IF YES
061516  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
061520  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
061524  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
061530  110337  061574                  MOVB    R3,8$           ;;SAVE FOR TYPING
061534  104401  061574                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
061540  105337  061576          7$:     DECB    $OCNT           ;;COUNT BY 1
061544  003347                          BGT     2$              ;;BR IF MORE TO DO
061546  002402                          BLT     6$              ;;BR IF DONE
061550  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
061552  000744                          BR      2$              ;;GO DO THE LAST DIGIT
061554  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
061556  012604                          MOV     (SP)+,R4        ;;RESTORE R4
061560  012603                          MOV     (SP)+,R3        ;;RESTORE R3
061562  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
061570  012616                          MOV     (SP)+,(SP)
061572  000002                          RTI                     ;;RETURN
061574     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
061575     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
061576     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
061577     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
061600  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
                                .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
                                     ;*

061602  105737  001173               $TYPE:    TSTB     $TPFLG           ;;IS THERE A TERMINAL?
061606  100002                                 BPL      1$               ;;BR IF YES
061610  000000                                 HALT                      ;;HALT HERE IF NO TERMINAL
061612  000430                                 BR       3$               ;;LEAVE
061614  010046                       1$:       MOV      R0,-(SP)         ;;SAVE R0
061616  017600  000002                         MOV      @2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
061622  122737  000001  001242                 CMPB     #APTENV,$ENV     ;;RUNNING IN APT MODE
061630  001011                                 BNE      62$              ;;NO,GO CHECK FOR APT CONSOLE
061632  132737  000100  001243                 BITB     #APTSPOOL,$ENVM  ;;SPOOL MESSAGE TO APT
061640  001405                                 BEQ      62$              ;;NO,GO CHECK FOR CONSOLE
061642  010037  061652                         MOV      R0,61$           ;;SETUP MESSAGE ADDRESS FOR APT
061646  004737  064672                         JSR      PC,$ATY3         ;;SPOOL MESSAGE TO APT
061652  000000                       61$:      .WORD    0                ;;MESSAGE ADDRESS
061654  132737  000040  001243       62$:      BITB     #APTCSUP,$ENVM   ;;APT CONSOLE SUPPRESSED
061662  001003                                 BNE      60$              ;;YES,SKIP TYPE OUT
061664  112046                       2$:       MOVB     (R0)+,-(SP)      ;;PUSH CHARACTER TO BE TYPED ONTO STACK
061666  001005                                 BNE      4$               ;;BR IF IT ISN'T THE TERMINATOR
061670  005726                                 TST      (SP)+            ;;IF TERMINATOR POP IT OFF THE STACK
061672  012600                       60$:      MOV      (SP)+,R0         ;;RESTORE R0
061674  062716  000002               3$:       ADD      #2,(SP)          ;;ADJUST RETURN PC
061700  000002                                 RTI                       ;;RETURN
061702  122716  000011               4$:       CMPB     #HT,(SP)         ;;BRANCH IF <HT>
061706  001430                                 BEQ      8$
061710  122716  000200                         CMPB     #CRLF,(SP)       ;;BRANCH IF NOT <CRLF>
061714  001006                                 BNE      5$
061716  005726                                 TST      (SP)+            ;;POP  <CR><LF> EQUIV
061720  104401                                 TYPE                      ;;TYPE A CR AND LF
061722  001217                                 $CRLF
061724  105037  062060                         CLRB     $CHARCNT         ;;CLEAR CHARACTER COUNT
061730  000755                                 BR       2$               ;;GET NEXT CHARACTER
061732  004737  062014               5$:       JSR      PC,$TYPEC        ;;GO TYPE THIS CHARACTER
061736  123726  001172               6$:       CMPB     $FILLC,(SP)+     ;;IS IT TIME FOR FILLER CHARS.?
061742  001350                                 BNE      2$               ;;IF NO GO GET NEXT CHAR.
061744  013746  001170                         MOV      $NULL,-(SP)      ;;GET # OF FILLER CHARS. NEEDED
                                                                         ;;AND THE NULL CHAR.
061750  105366  000001               7$:       DECB     1(SP)            ;;DOES A NULL NEED TO BE TYPED?
061754  002770                                 BLT      6$               ;;BR IF NO--GO POP THE NULL OFF OF STACK
061756  004737  062014                         JSR      PC,$TYPEC        ;;GO TYPE A NULL
061762  105337  062060                         DECB     $CHARCNT         ;;DO NOT COUNT AS A COUNT
061766  000770                                 BR       7$               ;;LOOP

                                     ;HORIZONTAL TAB PROCESSOR

061770  112716  000040               8$:       MOVB     #' ,(SP)         ;;REPLACE TAB WITH SPACE
061774  004737  062014               9$:       JSR      PC,$TYPEC        ;;TYPE A SPACE
062000  132737  000007  062060                 BITB     #7,$CHARCNT      ;;BRANCH IF NOT AT
062006  001372                                 BNE      9$               ;;TAB STOP
062010  005726                                 TST      (SP)+            ;;POP SPACE OFF STACK
062012  000724                                 BR       2$               ;;GET NEXT CHARACTER
062014  105777  117144               $TYPEC:   TSTB     @$TPS            ;;WAIT UNTIL PRINTER IS READY
```

```
062020  100375                          BPL     $TYPEC
062022  116677  000002  117136          MOVB    2(SP),@$TPB         ;;LOAD CHAR TO BE TYPED INTO DATA REG.
062030  122766  000015  000002          CMPB    #CR,2(SP)           ;;IS CHARACTER A CARRIAGE RETURN?
062036  001003                          BNE     1$                  ;;BRANCH IF NO
062040  105037  062060                  CLRB    $CHARCNT            ;;YES--CLEAR CHARACTER COUNT
062044  000406                          BR      $TYPEX              ;;EXIT
062046  122766  000012  000002  1$:     CMPB    #LF,2(SP)           ;;IS CHARACTER A LINE FEED?
062054  001402                          BEQ     $TYPEX              ;;BRANCH IF YES
062056  105227                          INCB    (PC)+               ;;COUNT THE CHARACTER
062060  000000                  $CHARCNT:.WORD  0                   ;;CHARACTER COUNT STORAGE
062062  000207                  $TYPEX: RTS     PC

                                .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<7:0>
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

062064                          $SCOPE:
062064  104410                          CKSWR                       ;;TEST  FOR CHANGE IN SOFT-SWR
062066  032777  040000  117060  1$:     BIT     #BIT14,@SWR         ;;LOOP ON PRESENT TEST?
062074  001131                          BNE     $OVER               ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
062076  000416                  $XTSTR: BR      6$                  ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
062100  013746  000004                  MOV     @#ERRVEC,-(SP)      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
062104  012737  062124  000004          MOV     #5$,@#ERRVEC        ;;SET FOR TIMEOUT
062112  005737  177060                  TST     @#177060            ;;TIME OUT ON XOR?
062116  012637  000004                  MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
062122  000500                          BR      $SVLAD              ;;GO TO THE NEXT TEST
062124  022626                  5$:     CMP     (SP)+,(SP)+         ;;CLEAR THE STACK AFTER A TIME OUT
062126  012637  000004                  MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
062132  000440                          BR      7$                  ;;LOOP ON THE PRESENT TEST
062134                          6$:;#####END OF CODE FOR THE XOR TESTER#####
062134  032777  000400  117012          BIT     #BIT08,@SWR         ;;LOOP ON SPEC. TEST?
062142  001421                          BEQ     2$                  ;;BR IF NO
062144  005046                          CLR     -(SP)               ;;CLEAR A TEMP. LOCATION
062146  117716  117002                  MOVB    @SWR,(SP)           ;;PICKUP THE DESIRED TEST NUMBER
062152  001414                          BEQ     8$                  ;;BRANCH IF BAD TEST NUMBER IN SWR
062154  022716  000121                  CMP     #121,(SP)           ;;CHECK THE NUMBER IN THE SWR
062160  002411                          BLT     8$                  ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
062162  011637  001116                  MOV     (SP),$TSTNM         ;;UPDATE THE TEST NUMBER
062166  005316                          DEC     (SP)                ;;BACKUP BY ONE
062170  006316                          ASL     (SP)                ;;SCALE THE TEST NUMBER AS AN INDEX
062172  062716  062376                  ADD     #$SW08TBL,(SP)      ;;FORM THE ADDRESS OF TEST POINTER
062176  013637  001122                  MOV     @(SP)+,$LPADR       ;;SET LOOP ADDRESS TO DESIRED TEST
062202  000466                          BR      $OVER               ;;GO LOOP ON THE TEST
062204  005726                  8$:     TST     (SP)+               ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
```

```
062206  105737  001117          2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
062212  001421                          BEQ     3$                      ;;BR IF NO
062214  123737  001131  001117          CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
062222  101015                          BHI     3$                      ;;BR IF NO
062224  032777  001000  116722          BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
062232  001404                          BEQ     4$                      ;;BR IF NO
062234  013737  001124  001122  7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
062242  000446                          BR      $OVER
062244  105037  001117          4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
062250  005037  001206                  CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
062254  000415                          BR      1$                      ;;ESCAPE TO THE NEXT TEST
062256  032777  004000  116670  3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
062264  001011                          BNE     1$                      ;;BR IF YES
062266  005737  001230                  TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
062272  001406                          BEQ     1$                      ;;      INHIBIT ITERATIONS
062274  005237  001120                  INC     $ICNT                   ;;INCREMENT ITERATION COUNT
062300  023737  001206  001120          CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
062306  002024                          BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
062310  012737  000001  001120  1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
062316  013737  062374  001206          MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
062324  105237  001116          $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
062330  113737  001116  001226          MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
062336  011637  001122                  MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
062342  011637  001124                  MOV     (SP),$LPERR             ;;SAVE ERROR LOOP ADDRESS
062346  005037  001210                  CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
062352  112737  000001  001131          MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
062360  013777  001116  116570  $OVER:  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
062366  013716  001122                  MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
062372  000002                          RTI                             ;;FIXES PS
062374  000012                  $MXCNT: 10.                             ;;MAX. NUMBER OF ITERATIONS
062376                          $SWOBTBL:
062376  006230                          .WORD   TST1+2                  ;;STARTING ADDRESS OF TEST 1
062400  006604                          .WORD   TST2+2                  ;;STARTING ADDRESS OF TEST 2
062402  007024                          .WORD   TST3+2                  ;;STARTING ADDRESS OF TEST 3
062404  007240                          .WORD   TST4+2                  ;;STARTING ADDRESS OF TEST 4
062406  007426                          .WORD   TST5+2                  ;;STARTING ADDRESS OF TEST 5
062410  010564                          .WORD   TST6+2                  ;;STARTING ADDRESS OF TEST 6
062412  011744                          .WORD   TST7+2                  ;;STARTING ADDRESS OF TEST 7
062414  012104                          .WORD   TST10+2                 ;;STARTING ADDRESS OF TEST 10
062416  012226                          .WORD   TST11+2                 ;;STARTING ADDRESS OF TEST 11
062420  012524                          .WORD   TST12+2                 ;;STARTING ADDRESS OF TEST 12
062422  013136                          .WORD   TST13+2                 ;;STARTING ADDRESS OF TEST 13
062424  013764                          .WORD   TST14+2                 ;;STARTING ADDRESS OF TEST 14
062426  014520                          .WORD   TST15+2                 ;;STARTING ADDRESS OF TEST 15
062430  014656                          .WORD   TST16+2                 ;;STARTING ADDRESS OF TEST 16
062432  015232                          .WORD   TST17+2                 ;;STARTING ADDRESS OF TEST 17
062434  015372                          .WORD   TST20+2                 ;;STARTING ADDRESS OF TEST 20
062436  015714                          .WORD   TST21+2                 ;;STARTING ADDRESS OF TEST 21
062440  016070                          .WORD   TST22+2                 ;;STARTING ADDRESS OF TEST 22
062442  016254                          .WORD   TST23+2                 ;;STARTING ADDRESS OF TEST 23
062444  016564                          .WORD   TST24+2                 ;;STARTING ADDRESS OF TEST 24
062446  017122                          .WORD   TST25+2                 ;;STARTING ADDRESS OF TEST 25
062450  017604                          .WORD   TST26+2                 ;;STARTING ADDRESS OF TEST 26
062452  017744                          .WORD   TST27+2                 ;;STARTING ADDRESS OF TEST 27
062454  020270                          .WORD   TST30+2                 ;;STARTING ADDRESS OF TEST 30
062456  020642                          .WORD   TST31+2                 ;;STARTING ADDRESS OF TEST 31
```

```
062460	021170		.WORD	TST32+2		;;STARTING ADDRESS OF TEST 32
062462	021744		.WORD	TST33+2		;;STARTING ADDRESS OF TEST 33
062464	022314		.WORD	TST34+2		;;STARTING ADDRESS OF TEST 34
062466	022644		.WORD	TST35+2		;;STARTING ADDRESS OF TEST 35
062470	023224		.WORD	TST36+2		;;STARTING ADDRESS OF TEST 36
062472	023612		.WORD	TST37+2		;;STARTING ADDRESS OF TEST 37
062474	024332		.WORD	TST40+2		;;STARTING ADDRESS OF TEST 40
062476	024662		.WORD	TST41+2		;;STARTING ADDRESS OF TEST 41
062500	025124		.WORD	TST42+2		;;STARTING ADDRESS OF TEST 42
062502	025564		.WORD	TST43+2		;;STARTING ADDRESS OF TEST 43
062504	025772		.WORD	TST44+2		;;STARTING ADDRESS OF TEST 44
062506	026410		.WORD	TST45+2		;;STARTING ADDRESS OF TEST 45
062510	027156		.WORD	TST46+2		;;STARTING ADDRESS OF TEST 46
062512	027510		.WORD	TST47+2		;;STARTING ADDRESS OF TEST 47
062514	030150		.WORD	TST50+2		;;STARTING ADDRESS OF TEST 50
062516	030442		.WORD	TST51+2		;;STARTING ADDRESS OF TEST 51
062520	030706		.WORD	TST52+2		;;STARTING ADDRESS OF TEST 52
062522	031576		.WORD	TST53+2		;;STARTING ADDRESS OF TEST 53
062524	032106		.WORD	TST54+2		;;STARTING ADDRESS OF TEST 54
062526	032340		.WORD	TST55+2		;;STARTING ADDRESS OF TEST 55
062530	032602		.WORD	TST56+2		;;STARTING ADDRESS OF TEST 56
062532	033034		.WORD	TST57+2		;;STARTING ADDRESS OF TEST 57
062534	033252		.WORD	TST60+2		;;STARTING ADDRESS OF TEST 60
062536	033526		.WORD	TST61+2		;;STARTING ADDRESS OF TEST 61
062540	033724		.WORD	TST62+2		;;STARTING ADDRESS OF TEST 62
062542	034200		.WORD	TST63+2		;;STARTING ADDRESS OF TEST 63
062544	034344		.WORD	TST64+2		;;STARTING ADDRESS OF TEST 64
062546	034546		.WORD	TST65+2		;;STARTING ADDRESS OF TEST 65
062550	035034		.WORD	TST66+2		;;STARTING ADDRESS OF TEST 66
062552	035302		.WORD	TST67+2		;;STARTING ADDRESS OF TEST 67
062554	035560		.WORD	TST70+2		;;STARTING ADDRESS OF TEST 70
062556	036046		.WORD	TST71+2		;;STARTING ADDRESS OF TEST 71
062560	036310		.WORD	TST72+2		;;STARTING ADDRESS OF TEST 72
062562	036552		.WORD	TST73+2		;;STARTING ADDRESS OF TEST 73
062564	037036		.WORD	TST74+2		;;STARTING ADDRESS OF TEST 74
062566	037202		.WORD	TST75+2		;;STARTING ADDRESS OF TEST 75
062570	037364		.WORD	TST76+2		;;STARTING ADDRESS OF TEST 76
062572	037544		.WORD	TST77+2		;;STARTING ADDRESS OF TEST 77
062574	037726		.WORD	TST100+2		;;STARTING ADDRESS OF TEST 100
062576	040110		.WORD	TST101+2		;;STARTING ADDRESS OF TEST 101
062600	040330		.WORD	TST102+2		;;STARTING ADDRESS OF TEST 102
062602	040570		.WORD	TST103+2		;;STARTING ADDRESS OF TEST 103
062604	040754		.WORD	TST104+2		;;STARTING ADDRESS OF TEST 104
062606	041216		.WORD	TST105+2		;;STARTING ADDRESS OF TEST 105
062610	041404		.WORD	TST106+2		;;STARTING ADDRESS OF TEST 106
062612	041636		.WORD	TST107+2		;;STARTING ADDRESS OF TEST 107
062614	042064		.WORD	TST110+2		;;STARTING ADDRESS OF TEST 110
062616	042372		.WORD	TST111+2		;;STARTING ADDRESS OF TEST 111
062620	042734		.WORD	TST112+2		;;STARTING ADDRESS OF TEST 112
062622	043206		.WORD	TST113+2		;;STARTING ADDRESS OF TEST 113
062624	044630		.WORD	TST114+2		;;STARTING ADDRESS OF TEST 114
062626	046406		.WORD	TST115+2		;;STARTING ADDRESS OF TEST 115
062630	050572		.WORD	TST116+2		;;STARTING ADDRESS OF TEST 116
062632	051202		.WORD	TST117+2		;;STARTING ADDRESS OF TEST 117
062634	053354		.WORD	TST120+2		;;STARTING ADDRESS OF TEST 120
062636	055672		.WORD	TST121+2		;;STARTING ADDRESS OF TEST 121
```

```
                                        .SBTTL  ERROR HANDLER ROUTINE

                                        ;;*****************************************************
                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO ERRTYP ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*       ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

 062640                                 $ERROR:
 062640  104410                                 CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 062642  105237  001117                 7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
 062646  001775                                 BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 062650  013777  001116  116300                 MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 062656  032777  002000  116270                 BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 062664  001402                                 BEQ     1$              ;;NO - SKIP
 062666  104401  001212                         TYPE    ,$BELL          ;;RING BELL
 062672  005237  001126                 1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 062676  011637  001132                         MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 062702  162737  000002  001132                 SUB     #2,$ERRPC
 062710  117737  116216  001130                 MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 062716  032777  020000  116230                 BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 062724  001004                                 BNE     20$             ;;SKIP TYPEOUTS
 062726  004737  057314                         JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
 062732  104401  001217                         TYPE    ,$CRLF
 062736                                 20$:
 062736  122737  000001  001242                 CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 062744  001007                                 BNE     2$              ;;NO,SKIP APT ERROR REPORT
 062746  113737  001130  062760                 MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 062754  004737  064702                         JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 062760     000                         21$:    .BYTE   0
 062761     000                                 .BYTE   0
 062762  000777                         22$:    BR      22$             ;;APT ERROR LOOP
 062764  005777  116164                 2$:     TST     @SWR            ;;HALT ON ERROR
 062770  100002                                 BPL     3$              ;;SKIP IF CONTINUE
 062772  000000                                 HALT                    ;;HALT ON ERROR!
 062774  104410                                 CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 062776  032777  001000  116150         3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 063004  001402                                 BEQ     4$              ;;BR IF NO
 063006  013716  001124                         MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 063012  005737  001210                 4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 063016  001402                                 BEQ     5$              ;;BR IF NONE
 063020  013716  001210                         MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 063024                                 5$:
 063024  022737  057274  000042                 CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 063032  001001                                 BNE     6$              ;;BRANCH IF NO
 063034  000000                                 HALT                    ;;YES
 063036                                 6$:
 063036  000002                                 RTI                     ;;RETURN
                                        .SBTTL  TTY INPUT ROUTINE
```

```
                                        ;;*****************************************************************
                                        .ENABL  LSB
063040  000000                          $TKCNT: .WORD   0                       ;;NUMBER OF ITEMS IN QUEUE
063042  000000                          $TKQIN: .WORD   0                       ;;INPUT POINTER
063044  000000                          $TKQOUT:  .WORD 0                       ;;OUTPUT POINTER
063046  000001                          $TKQSRT:  .BLKB 1                       ;;TTY KEYBOARD QUEUE
        063047                          $TKQEND=.
        063050                          .EVEN

                                        ;*TK INITIALIZE ROUTINE
                                        ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                        ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                        ;
                                        ;*CALL:
                                        ;*       JSR     PC,$TKINT
                                        ;*       RETURN

063050  005037  063040                  $TKINT: CLR     $TKCNT                  ;;CLEAR COUNT OF ITEMS IN QUEUE
063054  012737  063046  063042                  MOV     #$TKQSRT,$TKQIN         ;;MOVE THE STARTING ADDRESS OF THE
063062  013737  063042  063044                  MOV     $TKQIN,$TKQOUT          ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
063070  012737  063120  000060                  MOV     #$TKSRV,@#TKVEC         ;;INITIALIZE THE KEYBOARD VECTOR
063076  012737  000200  000062                  MOV     #200,@#TKVEC+2          ;;"BR" LEVEL 4
063104  005777  116052                          TST     @$TKB                   ;;CLEAR DONE FLAG
063110  012777  000100  116042                  MOV     #100,@$TKS              ;;ENABLE TTY KEYBOARD INTERRUPT
063116  000207                                  RTS     PC                      ;;RETURN TO CALLER

                                        ;*TK SERVICE ROUTINE
                                        ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                        ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                        ;*IT IN THE QUEUE.
                                        ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
                                        ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (START)
                                        ;
063120  117746  116036                  $TKSRV: MOVB    @$TKB,-(SP)             ;;PICKUP THE CHARACTER
063124  042716  177600                          BIC     #↑C177,(SP)             ;;STRIP THE JUNK
063130  021627  000003                          CMP     (SP),#3                 ;;IS IT A CONTROL C?
063134  001007                                  BNE     1$                      ;;BRANCH IF NO
063136  104401  064234                          TYPE    ,$CNTLC                 ;;TYPE A CONTROL-C (↑C)
063142  004737  063050                          JSR     PC,$TKINT               ;;INIT THE KEYBOARD
063146  005726                                  TST     (SP)+                   ;;CLEAN UP STACK
063150  000137  004542                          JMP     START                   ;;CONTROL C RESTART
063154  021627  000007                  1$:     CMP     (SP),#7                 ;;IS IT A CONTROL G?
063160  001004                                  BNE     2$                      ;;BRANCH IF NO
063162  022737  000176  001154                  CMP     #SWREG,SWR              ;;IS SOFT-SWR SELECTED?
063170  001500                                  BEQ     6$                      ;;GO TO SWR CHANGE

063172                                  2$:
063172  022737  000001  063040                  CMP     #1,$TKCNT               ;;IS THE QUEUE FULL?
063200  001004                                  BNE     3$                      ;;BRANCH IF NO
063202  104401  001212                          TYPE    ,$BELL                  ;;RING THE TTY BELL
063206  005726                                  TST     (SP)+                   ;;CLEAN CHARACTER OFF OF STACK
063210  000451                                  BR      5$                      ;;EXIT
063212  021627  000023                  3$:     CMP     (SP),#23                ;;IS IT A CONTROL-S?
063216  001021                                  BNE     32$                     ;;BRANCH IF NO
063220  005077  115734                          CLR     @$TKS                   ;;DISABLE TTY KEYBOARD INTERRUPTS
063224  005726                                  TST     (SP)+                   ;;CLEAN CHAR OFF STACK
```

```
063226  105777  115726          31$:    TSTB    @$TKS                   ;;WAIT FOR A CHAR
063232  100375                          BPL     31$                     ;;LOOP UNTIL ITS THERE
063234  117746  115722                  MOVB    @$TKB,-(SP)             ;;GET THE CHARACTER
063240  042716  177600                  BIC     #↑C177,(SP)             ;;MAKE IT 7-BIT ASCII
063244  022627  000021                  CMP     (SP)+,#21               ;;IS IT A CONTROL-Q?
063250  001366                          BNE     31$                     ;;BRANCH IF NO
063252  012777  000100  115700          MOV     #100,@$TKS              ;;REENABLE TTY KEYBOARD INTERRUPTS
063260  000002                          RTI                             ;;RETURN
063262  005237  063040          32$:    INC     $TKCNT                  ;;COUNT THIS CHARACTER
063266  021627  000140                  CMP     (SP),#140               ;;IS IT UPPER CASE?
063272  002405                          BLT     4$                      ;;BRANCH IF YES
063274  021627  000175                  CMP     (SP),#175               ;;IS IT A SPECIAL CHAR?
063300  003002                          BGT     4$                      ;;BRANCH IF YES
063302  042716  000040                  BIC     #40,(SP)                ;;MAKE IT UPPER CASE
063306  112677  177530          4$:     MOVB    (SP)+,@$TKQIN           ;;AND PUT IT IN QUEUE
063312  005237  063042                  INC     $TKQIN                  ;;UPDATE THE POINTER
063316  023727  063042  063047          CMP     $TKQIN,#$TKQEND         ;;GO OFF THE END?
063324  001003                          BNE     5$                      ;;BRANCH IF NO
063326  012737  063046  063042          MOV     #$TKQSRT,$TKQIN         ;;RESET THE POINTER
063334  000002                  5$:     RTI                             ;;RETURN

                                ;;**************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                                ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
063336  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR              ;;IS THE SOFT-SWR SELECTED
063344  001124                          BNE     15$                     ;;EXIT IF NOT
063346  105777  115606                  TSTB    @$TKS                   ;;IS A CHAR WAITING?
063352  100121                          BPL     15$                     ;;IF NOT, EXIT
063354  117746  115602                  MOVB    @$TKB,-(SP)             ;;YES
063360  042716  177600                  BIC     #↑C177,(SP)             ;;MAKE IT  7-BIT ASCII
063364  021627  000007                  CMP     (SP),#7                 ;;IS IT A CONTROL-G?
063370  001300                          BNE     2$                      ;;IF NOT, PUT IT IN THE TTY QUEUE
                                                                        ;;AND EXIT

                                ;;**************************************************************
                                ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
                                ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
                                ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
063372  123727  001150  000001  6$:     CMPB    $AUTOB,#1               ;;ARE WE RUNNING IN AUTO-MODE?
063400  001674                          BEQ     2$                      ;;BRANCH IF YES
063402  005726                          TST     (SP)+                   ;;CLEAR CONTROL-G OFF STACK
063404  004737  063050                  JSR     PC,$TKINT               ;;FLUSH THE TTY INPUT QUEUE
063410  005077  115544                  CLR     @$TKS                   ;;DISABLE TTY KEYBOARD INTERRUPTS
063414  112737  000001  001151          MOVB    #1,$INTAG               ;;SET INTERRUPT MODE INDICATOR

063422  104401  064246                  TYPE    ,$CNTLG                 ;;ECHO THE CONTROL-G (↑G)
063426  104401  064253          $GTSWR: TYPE    ,$MSWR                  ;;TYPE CURRENT CONTENTS
063432  013746  000176                  MOV     SWREG,-(SP)             ;;SAVE SWREG FOR TYPEOUT
063436  104402                          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
063440  104401  064264                  TYPE    ,$MNEW                  ;;PROMPT FOR NEW SWR
063444  005046                  19$:    CLR     -(SP)                   ;;CLEAR COUNTER
063446  005046                          CLR     -(SP)                   ;;THE NEW SWR
063450  105777  115504          7$:     TSTB    @$TKS                   ;;CHAR THERE?
063454  100375                          BPL     7$                      ;;IF NOT TRY AGAIN
```

```

063456  117746  115500                   MOVB    @$TKB,-(SP)          ;;PICK UP CHAR
063462  042716  177600                   BIC     #^C177,(SP)          ;;MAKE IT 7-BIT ASCII

063466  021627  000003                   CMP     (SP),#3              ;;IS IT A CONTROL-C?
063472  001015                           BNE     9$                   ;;BRANCH IF NOT
063474  104401  064234                   TYPE    ,$CNTLC              ;;YES, ECHO CONTROL-C (^C)
063500  062706  000006                   ADD     #6,SP                ;;CLEAN UP STACK
063504  123727  001151  000001           CMPB    $INTAG,#1            ;;REENABLE TTY KEYBOARD INTERRUPTS?
063512  001003                           BNE     8$                   ;;BRANCH IF NO
063514  012777  000100  115436           MOV     #100,@$TKS           ;;ALLOW TTY KEYBOARD INTERRUPTS
063522  000137  004542          8$:      JMP     START                ;;CONTROL-C RESTART


063526  021627  000025          9$:      CMP     (SP),#25             ;;IS IT A CONTROL-U?
063532  001005                           BNE     10$                  ;;BRANCH IF NOT
063534  104401  064241                   TYPE    ,$CNTLU              ;;YES, ECHO CONTROL-U (^U)
063540  062706  000006          20$:     ADD     #6,SP                ;;IGNORE PREVIOUS INPUT
063544  000737                           BR      19$                  ;;LET'S TRY IT AGAIN


063546  021627  000015          10$:     CMP     (SP),#15             ;;IS IT A <CR>?
063552  001022                           BNE     16$                  ;;BRANCH IF NO
063554  005766  000004                   TST     4(SP)                ;;YES, IS IT THE FIRST CHAR?
063560  001403                           BEQ     11$                  ;;BRANCH IF YES
063562  016677  000002  115364           MOV     2(SP),@SWR           ;;SAVE NEW SWR
063570  062706  000006          11$:     ADD     #6,SP                ;;CLEAR UP STACK
063574  104401  001217          14$:     TYPE    ,$CRLF               ;;ECHO <CR> AND <LF>
063600  123727  001151  000001           CMPB    $INTAG,#1            ;;RE-ENABLE TTY KBD INTERRUPTS?
063606  001003                           BNE     15$                  ;;BRANCH IF NOT
063610  012777  000100  115342           MOV     #100,@$TKS           ;;RE-ENABLE TTY KBD INTERRUPTS
063616  000002                  15$:     RTI                          ;;RETURN
063620  004737  062014          16$:     JSR     PC,$TYPEC            ;;ECHO CHAR
063624  021627  000060                   CMP     (SP),#60             ;;CHAR < 0?
063630  002420                           BLT     18$                  ;;BRANCH IF YES
063632  021627  000067                   CMP     (SP),#67             ;;CHAR > 7?
063636  003015                           BGT     18$                  ;;BRANCH IF YES
063640  042726  000060                   BIC     #60,(SP)+            ;;STRIP-OFF ASCII
063644  005766  000002                   TST     2(SP)                ;;IS THIS THE FIRST CHAR
063650  001403                           BEQ     17$                  ;;BRANCH IF YES
063652  006316                           ASL     (SP)                 ;;NO, SHIFT PRESENT
063654  006316                           ASL     (SP)                 ;;    CHAR OVER TO MAKE
063656  006316                           ASL     (SP)                 ;;    ROOM FOR NEW ONE.
063660  005266  000002          17$:     INC     2(SP)                ;;KEEP COUNT OF CHAR
063664  056616  177776                   BIS     -2(SP),(SP)          ;;SET IN NEW CHAR
063670  000667                           BR      7$                   ;;GET THE NEXT ONE
063672  104401  001216          18$:     TYPE    ,$QUES               ;;TYPE ?<CR><LF>
063676  000720                           BR      20$                  ;;SIMULATE CONTROL-U
                                .DSABL   LSB



                                ;;*****************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*        RDCHR                       ;;GET A CHARACTER FROM THE QUEUE
                                ;*        RETURN HERE                 ;;CHARACTER IS ON THE STACK
```

```
                                        ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                        ;
063700  011646                          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
063702  016666  000004  000002                  MOV     4(SP),2(SP)     ;;THE PS
063710  005066  000004                          CLR     4(SP)           ;;GET READY FOR A CHARACTER
063714  005046                                  CLR     -(SP)           ;;PUT NEW PS ON STACK
063716  012746  063724                          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
063722  000002                                  RTI                     ;;POP NEW PC AND PS
063724                                  64$:
063724  005737  063040                  1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
063730  001775                                  BEQ     1$
063732  005337  063040                          DEC     $TKCNT          ;;DECREMENT THE COUNTER
063736  117766  177102  000004                  MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
063744  005237  063044                          INC     $TKQOUT         ;;UPDATE THE POINTER
063750  023727  063044  063047                  CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
063756  001003                                  BNE     2$              ;;BRANCH IF NO
063760  012737  063046  063044                  MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
063766  000002                          2$:     RTI                     ;;RETURN

                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                        ;*CALL:
                                        ;*      RDLIN                           ;;INPUT A STRING FROM THE TTY
                                        ;*      RETURN HERE                     ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                        ;*                                      ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

063770  010346                          $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
063772  005046                                  CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
063774  012703  064224                  1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
064000  022703  064234                  2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
064004  101456                                  BLOS    4$              ;;BR IF YES
064006  104411                                  RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
064010  112613                                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
064012  122713  000177                  10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
064016  001022                                  BNE     5$              ;;BR IF NO
064020  005716                                  TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
064022  001007                                  BNE     6$              ;;BR IF NO
064024  112737  000134  064222                  MOVB    #'\,9$          ;;TYPE A BACK SLASH
064032  104401  064222                          TYPE    ,9$
064036  012716  177777                          MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
064042  005303                          6$:     DEC     R3              ;;BACKUP BY ONE
064044  020327  064224                          CMP     R3,#$TTYIN      ;;STACK EMPTY?
064050  103434                                  BLO     4$              ;;BR IF YES
064052  111337  064222                          MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
064056  104401  064222                          TYPE    ,9$             ;;GO TYPE
064062  000746                                  BR      2$              ;;GO READ ANOTHER CHAR.
064064  005716                          5$:     TST     (SP)            ;;RUBOUT KEY SET?
064066  001406                                  BEQ     7$              ;;BR IF NO
064070  112737  000134  064222                  MOVB    #'\,9$          ;;TYPE A BACK SLASH
064076  104401  064222                          TYPE    ,9$
064102  005016                                  CLR     (SP)            ;;CLEAR THE RUBOUT KEY
064104  122713  000025                  7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
064110  001003                                  BNE     8$              ;;BR IF NO
064112  104401  064241                          TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
064116  000726                                  BR      1$              ;;GO START OVER
064120  122713  000022                  8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
```

```
064124  001011                             BNE     3$                      ;;BRANCH IF NO
064126  105013                             CLRB    (R3)                    ;;CLEAR THE CHARACTER
064130  104401  001217                     TYPE    ,$CRLF                  ;;TYPE A "CR" & "LF"
064134  104401  064224                     TYPE    ,$TTYIN                 ;;TYPE THE INPUT STRING
064140  000717                             BR      2$                      ;;GO PICKUP ANOTHER CHACTER
064142  104401  001216             4$:     TYPE    ,$QUES                  ;;TYPE A '?'
064146  000712                             BR      1$                      ;;CLEAR THE BUFFER AND LOOP
064150  111337  064222             3$:     MOVB    (R3),9$                 ;;ECHO THE CHARACTER
064154  104401  064222                     TYPE    ,9$
064160  122723  000015                     CMPB    #15,(R3)+               ;;CHECK FOR RETURN
064164  001305                             BNE     2$                      ;;LOOP IF NOT RETURN
064166  105063  177777                     CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
064172  104401  001220                     TYPE    ,$LF                    ;;TYPE A LINE FEED
064176  005726                             TST     (SP)+                   ;;CLEAN RUBOUT KEY FROM THE STACK
064200  012603                             MOV     (SP)+,R3                ;;RESTORE R3
064202  011646                             MOV     (SP),-(SP)              ;;ADJUST THE STACK AND PUT ADDRESS OF THE
064204  016666  000004  000002             MOV     4(SP),2(SP)             ;;        FIRST ASCII CHARACTER ON IT
064212  012766  064224  000004             MOV     #$TTYIN,4(SP)
064220  000002                             RTI                             ;;RETURN
064222     000                     9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
064223     000                             .BYTE   0                       ;;TERMINATOR
064224  000010                     $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
064234  041536  005015     000     $CNTLC: .ASCIZ  /↑C/<15><12>            ;;CONTROL "C"
064241     136  006525  000012     $CNTLU: .ASCIZ  /↑U/<15><12>            ;;CONTROL "U"
064246  043536  005015     000     $CNTLG: .ASCIZ  /↑G/<15><12>            ;;CONTROL "G"
064253     015  051412  051127     $MSWR:  .ASCIZ  <15><12>/SWR = /
064260  036440  000040
064264  020040  042516  020127     $MNEW:  .ASCIZ  /  NEW = /
064272  020075     000
        064276                     .EVEN
                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                   ;;*********************************************************
                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                   ;*CHANGE IT TO BINARY.
                                   ;*CALL:
                                   ;*      RDOCT                           ;;READ AN OCTAL NUMBER
                                   ;*      RETURN HERE                     ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                   ;*                                      ;;HIGH ORDER BITS ARE IN $HIOCT

064276  011646                     $RDOCT: MOV     (SP),-(SP)              ;;PROVIDE SPACE FOR THE
064300  016666  000004  000002             MOV     4(SP),2(SP)             ;;INPUT NUMBER
064306  010046                             MOV     R0,-(SP)                ;;PUSH R0 ON STACK
064310  010146                             MOV     R1,-(SP)                ;;PUSH R1 ON STACK
064312  010246                             MOV     R2,-(SP)                ;;PUSH R2 ON STACK
064314  104412                     1$:     RDLIN                           ;;READ AN ASCIZ LINE
064316  012600                             MOV     (SP)+,R0                ;;GET ADDRESS OF 1ST CHARACTER
064320  005001                             CLR     R1                      ;;CLEAR DATA WORD
064322  005002                             CLR     R2
064324  112046                     2$:     MOVB    (R0)+,-(SP)             ;;PICKUP THIS CHARACTER
064326  001412                             BEQ     3$                      ;;IF ZERO GET OUT
064330  006301                             ASL     R1                      ;;*2
064332  006102                             ROL     R2
064334  006301                             ASL     R1                      ;;*4
064336  006102                             ROL     R2
064340  006301                             ASL     R1                      ;;*8
```

```
064342  006102                     ROL     R2
064344  042716  177770             BIC     #^C7,(SP)          ;;STRIP THE ASCII JUNK
064350  062601                     ADD     (SP)+,R1           ;;ADD IN THIS DIGIT
064352  000764                     BR      2$                 ;;LOOP
064354  005726             3$:     TST     (SP)+              ;;CLEAN TERMINATOR FROM STACK
064356  010166  000012             MOV     R1,12(SP)          ;;SAVE THE RESULT
064362  010237  064376             MOV     R2,$HIOCT
064366  012602                     MOV     (SP)+,R2           ;;POP STACK INTO R2
064370  012601                     MOV     (SP)+,R1           ;;POP STACK INTO R1
064372  012600                     MOV     (SP)+,R0           ;;POP STACK INTO R0
064374  000002                     RTI                        ;;RETURN
064376  000000             $HIOCT: .WORD   0                  ;;HIGH ORDER BITS GO HERE
                           .SBTTL  TRAP DECODER


                           ;;*******************************************************
                           ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                           ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                           ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                           ;*GO TO THAT ROUTINE.

064400  016646  000002     $TRAP:  MOV     2(SP),-(SP)        ;;ASSUME THE STATUS OF
064404  042716  000020             BIC     #20,(SP)           ;; THE CALLER--DO NOT ALLOW
064410  012746  064416             MOV     #1$,-(SP)          ;; T-BIT TRAPS
064414  000002                     RTI                        ;;SET THE NEW STATUS
064416  010046             1$:     MOV     R0,-(SP)           ;;SAVE R0
064420  016600  000002             MOV     2(SP),R0           ;;GET TRAP ADDRESS
064424  005740                     TST     -(R0)              ;;BACKUP BY 2
064426  111000                     MOVB    (R0),R0            ;;GET RIGHT BYTE OF TRAP
064430  006300                     ASL     R0                 ;;POSITION FOR INDEXING
064432  016000  064452             MOV     $TRPAD(R0),R0      ;;INDEX TO TABLE
064436  000200                     RTS     R0                 ;;GO TO ROUTINE


                           ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

064440  011646             $TRAP2: MOV     (SP),-(SP)         ;;MOVE THE PC DOWN
064442  016666  000004  000002     MOV     4(SP),2(SP)        ;;MOVE THE PSW DOWN
064450  000002                     RTI                        ;;RESTORE THE PSW

                           .SBTTL  TRAP TABLE

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

                           ;       ROUTINE
                           ;       -------
064452  064440             $TRPAD: .WORD   $TRAP2
064454  061602                     $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
064456  061400                     $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
064460  061354                     $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
064462  061414                     $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
064464  061130                     $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
064466  061054                     $TYPBN  ;;CALL=TYPBN    TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER

064470  063426                     $GTSWR  ;;CALL=GTSWR    TRAP+7(104407)  GET SOFT-SWR SETTING

```

```
064472  063336                          $CKSWR  ;;CALL=CKSWR      TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
064474  063700                          $RDCHR  ;;CALL=RDCHR      TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
064476  063770                          $RDLIN  ;;CALL=RDLIN      TRAP+12(104412) TTY TYPEIN STRING ROUTINE
064500  064276                          $RDOCT  ;;CALL=RDOCT      TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
064502  060760                          $SAVREG ;;CALL=SAVREG     TRAP+14(104414) SAVE R0-R5 ROUTINE
064504  061016                          $RESREG ;;CALL=RESREG     TRAP+15(104415) RESTORE R0-R5 ROUTINE
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
064506  012737  064646  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC  ;;SET FOR FAST UP
064514  012737  000340  000026          MOV     #340,@#PWRVEC+2   ;;PRIO:7
064522  010046                          MOV     R0,-(SP)          ;;PUSH R0 ON STACK
064524  010146                          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
064526  010246                          MOV     R2,-(SP)          ;;PUSH R2 ON STACK
064530  010346                          MOV     R3,-(SP)          ;;PUSH R3 ON STACK
064532  010446                          MOV     R4,-(SP)          ;;PUSH R4 ON STACK
064534  010546                          MOV     R5,-(SP)          ;;PUSH R5 ON STACK
064536  017746  114412                  MOV     @SWR,-(SP)        ;;PUSH @SWR ON STACK
064542  010637  064652                  MOV     SP,$SAVR6         ;;SAVE SP
064546  012737  064560  000024          MOV     #$PWRUP,@#PWRVEC  ;;SET UP VECTOR
064554  000000                          HALT
064556  000776                          BR      .-2               ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
064560  012737  064646  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC  ;;SET FOR FAST DOWN
064566  013706  064652                  MOV     $SAVR6,SP         ;;GET SP
064572  005037  064652                  CLR     $SAVR6            ;;WAIT LOOP FOR THE TTY
064576  005237  064652          1$:     INC     $SAVR6            ;;WAIT FOR THE INC
064602  001375                          BNE     1$                ;;OF WORD
064604  012677  114344                  MOV     (SP)+,@SWR        ;;POP STACK INTO @SWR
064610  012605                          MOV     (SP)+,R5          ;;POP STACK INTO R5
064612  012604                          MOV     (SP)+,R4          ;;POP STACK INTO R4
064614  012603                          MOV     (SP)+,R3          ;;POP STACK INTO R3
064616  012602                          MOV     (SP)+,R2          ;;POP STACK INTO R2
064620  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
064622  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
064624  012737  064506  000024          MOV     #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
064632  012737  000340  000026          MOV     #340,@#PWRVEC+2   ;;PRIO:7
064640  104401                          TYPE                      ;REPORT THE POWER FAILURE
064642  064654                  $PWRMG: .WORD   $POWER            ;;POWER FAIL MESSAGE POINTER
064644  000002                          RTI
064646  000000                  $ILLUP: HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
064650  000776                          BR      .-2               ;; BEFORE THE POWER DOWN WAS COMPLETE
064652  000000                  $SAVR6: 0                         ;;PUT THE SP HERE
064654  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
064662  000122

                                        .EVEN
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;*****************************************************************
064664  112737  000001  065130  $ATY1:  MOVB    #1,$FFLG          ;;TO REPORT FATAL ERROR
064672  112737  000001  065126  $ATY3:  MOVB    #1,$MFLG          ;;TO TYPE A MESSAGE
064700  000403                          BR      $ATYC
064702  112737  000001  065130  $ATY4:  MOVB    #1,$FFLG          ;;TO ONLY REPORT FATAL ERROR
```

```
064710                                  $ATYC:
064710  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
064712  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
064714  105737  065126                          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
064720  001450                                  BEQ     5$              ;;IF NOT:  BR
064722  122737  000001  001242                  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
064730  001031                                  BNE     3$              ;;IF NOT:  BR
064732  132737  000100  001243                  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
064740  001425                                  BEQ     3$              ;;IF NOT:  BR
064742  017600  000004                          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
064746  062766  000002  000004                  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
064754  005737  001222          1$:             TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
064760  001375                                  BNE     1$              ;;IF NOT:  WAIT
064762  010037  001236                          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
064766  105720                  2$:             TSTB    (R0)+           ;;FIND END OF MESSAGE
064770  001376                                  BNE     2$
064772  163700  001236                          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
064776  006200                                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
065000  010037  001240                          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
065004  012737  000004  001222                  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
065012  000413                                  BR      5$
065014  017637  000004  065040  3$:             MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
065022  062766  000002  000004                  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
065030  013746  177776                          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
065034  004737  061602                          JSR     PC,$TYPE        ;;CALL TYPE MACRO
065040  000000                  4$:             .WORD   0
065042                          5$:
065042  105737  065130          10$:            TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
065046  001416                                  BEQ     12$             ;;IF NOT:  BR
065050  005737  001242                          TST     $ENV            ;;RUNNING UNDER APT?
065054  001413                                  BEQ     12$             ;;IF NOT:  BR
065056  005737  001222          11$:            TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
065062  001375                                  BNE     11$             ;;IF NOT:  WAIT
065064  017637  000004  001224                  MOV     @4(SP),$FATAL   ;;GET ERROR #
065072  062766  000002  000004                  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
065100  005237  001222                          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
065104  105037  065130          12$:            CLRB    $FFLG           ;;CLEAR FATAL FLAG
065110  105037  065127                          CLRB    $LFLG           ;;CLEAR LOG FLAG
065114  105037  065126                          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
065120  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
065122  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
065124  000207                                  RTS     PC              ;;RETURN
065126     000                  $MFLG:          .BYTE   0               ;;MESSG. FLAG
065127     000                  $LFLG:          .BYTE   0               ;;LOG FLAG
065130     000                  $FFLG:          .BYTE   0               ;;FATAL FLAG
        065132                                  .EVEN
        000200                  APTSIZE=200
        000001                  APTENV=001
        000100                  APTSPOOL=100
        000040                  APTCSUP=040
```

```
                                         .SBTTL  CONSOLE MESSAGES

065132      051                          CLSPRN: .ASCII  @)@
065133      075     000                  EQUALS: .ASCIZ  @=@
065135      015  025012     000          PROMPT: .ASCIZ  <CR><LF>@*@
065141      077     000                  QSTMRK: .ASCIZ  @?@
065143                                   HELPQST:
065143      015  052012  050131          .ASCIZ  <CR><LF>@TYPE HELP TEXT (Y OR N)??@
065150   020105  042510  050114
065156   052040  054105  020124
065164   054450  047440  020122
065172   024516  037477     000
065177                                   UBUSQST:
065177      015  041412  040510          .ASCIZ  <CR><LF>@CHANGE RM03 UNIBUS ADDRESS OR VECTOR ADDRESS (Y OR N) ??@
065204   043516  020105  046522
065212   031460  052440  044516
065220   052502  020123  042101
065226   051104  051505  020123
065234   051117  053040  041505
065242   047524  020122  042101
065250   051104  051505  020123
065256   054450  047440  020122
065264   024516  037440  000077
065272   005015  051525  020105          CNSL00: .ASCIZ  <CR><LF>@USE SAME DEVICES (Y OR N) ??@
065300   040523  042515  042040
065306   053105  041511  051505
065314   024040  020131  051117
065322   047040  020051  037477
065330      000
065331      015  051012  030115          CNSL01: .ASCIZ  <CR><LF>@RM03 BUS ADDRESS (@
065336   020063  052502  020123
065344   042101  051104  051505
065352   020123  000050
065356   005015  047105  051124          CNSL02: .ASCII  <CR><LF>@ENTRY NOT IN I/O PAGE@
065364   020131  047516  020124
065372   047111  044440  047457
065400   050040  043501     105
065405      015  040412  042104                  .ASCIZ  <CR><LF>@ADDRESS MUST BE >160000@
065412   042522  051523  046440
065420   051525  020124  042502
065426   037040  033061  030060
065434   030060     000
065437      015  051012  030115          CNSL03: .ASCIZ  <CR><LF>@RM03 VECTOR ADDRESS (@
065444   020063  042526  052103
065452   051117  040440  042104
065460   042522  051523  024040
065466      000
065467      015  042412  052116          CNSL04: .ASCII  <CR><LF>@ENTRY OUT OF RANGE@
065474   054522  047440  052125
065502   047440  020106  040522
065510   043516     105
065513      015  040412  042104                  .ASCIZ  <CR><LF>@ADDRESS MUST BE <1000@
065520   042522  051523  046440
065526   051525  020124  042502
065534   036040  030061  030060
065542      000
```

```
 065543     015  051012  030115  CNSL05: .ASCIZ  <CR><LF>@RM03 INTERRUPT PRIORITY (@
 065550  020063  047111  042524
 065556  051122  050125  020124
 065564  051120  047511  044522
 065572  054524  024040     000
 065577     015  042412  052116  CNSL06: .ASCIZ  <CR><LF>@ENTRY OUT OF RANGE@
 065604  054522  047440  052125
 065612  047440  020106  040522
 065620  043516  000105
 065624                          CNSL07:
 065624  005015  054524  042520  .ASCII  <CR><LF>@TYPE (A) TO TEST ALL DEVICES, OR TYPE DEVICE@
 065632  024040  024501  052040
 065640  020117  042524  052123
 065646  040440  046114  042040
 065654  053105  041511  051505
 065662  020054  051117  052040
 065670  050131  020105  042504
 065676  044526  042503
 065702  047040  046525  042502  .ASCII  @ NUMBER(S)@
 065710  024122  024523
 065714  005015  042524  046522  .ASCIZ  <CR><LF>@TERMINATE INPUT WITH CARRIAGE RETURN@
 065722  047111  052101  020105
 065730  047111  052520  020124
 065736  044527  044124  041440
 065744  051101  044522  043501
 065752  020105  042522  052524
 065760  047122     000

         065764                  .EVEN
```

```
        065764                  FNCDTB:                                 ;FUNCTION CODE TABLE

        065764  020000          .WORD   OPI                             ;NOP
        065766  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (2)
        065770  132000          .WORD   ATA!OPI!IVC!IAE                 ;SEEK
        065772  130000          .WORD   ATA!OPI!IVC                     ;RECALIBRATE
        065774  020000          .WORD   OPI                             ;DRIVE CLEAR
        065776  030000          .WORD   OPI!IVC                         ;RELEASE
        066000  130000          .WORD   OPI!ATA!IVC                     ;OFFSET
        066002  130000          .WORD   OPI!ATA!IVC                     ;RETURN TO CENTERLINE
        066004  020000          .WORD   OPI                             ;READ IN PRESET
        066006  020000          .WORD   OPI                             ;PACK ACKNOWLEDGE
        066010  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (24)
        066012  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (26)
        066014  132000          .WORD   ATA!OPI!IVC!IAE                 ;SEARCH
        066016  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (32)
        066020  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (34)
        066022  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (36)
        066024  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (40)
        066026  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (42)
        066030  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (44)
        066032  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (46)
        066034  073300          .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK DATA
        066036  073300          .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK HEADER AND DATA
        066040  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (54)
        066042  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (56)
        066044  037200          .WORD   OPI!IVC!WLE!IAE!AOE!HCE         ;WRITE DATA
        066046  037000          .WORD   OPI!IVC!WLE!IAE!AOE             ;WRITE HEADER AND DATA
        066050  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (64)
        066052  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (66)
        066054  033300          .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ DATA
        066056  033300          .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ HEADER AND DATA
        066060  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (74)
        066062  130001          .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (76)

        066064     001          ATNTBL: .BYTE   1.
        066065     002                  .BYTE   2.
        066066     004                  .BYTE   4.
        066067     010                  .BYTE   8.
        066070     020                  .BYTE   16.
        066071     040                  .BYTE   32.
        066072     100                  .BYTE   64.
        066073     200                  .BYTE   128.

        066074                  RGDTPT:
        066074  000000                  .WORD   0.
        066076  000001                  .WORD   1.
        066100  000003                  .WORD   3.
        066102  000007                  .WORD   7.
        066104  000017                  .WORD   15.
        066106  000037                  .WORD   31.
        066110  000077                  .WORD   63.
        066112  000177                  .WORD   127.
        066114  000377                  .WORD   255.
        066116  000777                  .WORD   511.
```

```
15113 066120 001777              .WORD  1023.
15114 066122 003777              .WORD  2047.
15115 066124 007777              .WORD  4095.
15116 066126 017777              .WORD  8191.
15117 066130 037777              .WORD  16383.
15118 066132 077777              .WORD  32767.
15119 066134 177777     ONES:    .WORD  65535.
15120 066136 177777              .WORD  65535.
15121 066140 077777              .WORD  32767.
15122 066142 037777              .WORD  16383.
15123 066144 017777              .WORD  8191.
15124 066146 007777              .WORD  4095.
15125 066150 003777              .WORD  2047.
15126 066152 001777              .WORD  1023.
15127 066154 000777              .WORD  511.
15128 066156 000377              .WORD  255.
15129 066160 000177              .WORD  127.
15130 066162 000077              .WORD  63.
15131 066164 000037              .WORD  31.
15132 066166 000017              .WORD  15.
15133 066170 000007              .WORD  7.
15134 066172 000003              .WORD  3.
15135 066174 000001              .WORD  1.
15136 066176 000000     ZEROS:   .WORD  0.
15137 066200 000000              .WORD  0.
15138 066202 000001              .WORD  1.
15139 066204 000002              .WORD  2.
15140 066206 000004              .WORD  4.
15141 066210 000010              .WORD  8.
15142 066212 000020              .WORD  16.
15143 066214 000040              .WORD  32.
15144 066216 000100              .WORD  64.
15145 066220 000200              .WORD  128.
15146 066222 000400              .WORD  256.
15147 066224 001000              .WORD  512.
15148 066226 002000              .WORD  1024.
15149 066230 004000              .WORD  2048.
15150 066232 010000              .WORD  4096.
15151 066234 020000              .WORD  8192.
15152 066236 040000              .WORD  16384.
15153 066240 100000              .WORD  32768.
15154 066242 100000              .WORD  32768.
15155 066244 040000              .WORD  16384.
15156 066246 020000              .WORD  8192.
15157 066250 010000              .WORD  4096.
15158 066252 004000              .WORD  2048.
15159 066254 002000              .WORD  1024.
15160 066256 001000              .WORD  512.
15161 066260 000400              .WORD  256.
15162 066262 000200              .WORD  128.
15163 066264 000100              .WORD  64.
15164 066266 000040              .WORD  32.
15165 066270 000020              .WORD  16.
15166 066272 000010              .WORD  8.
15167 066274 000004              .WORD  4.
15168 066276 000002              .WORD  2.
```

F08


```
15169  066300  000001          .WORD   1.
15170  066302  000000          .WORD   0.
15171  066304  177777          .WORD   65535.
15172  066306  177776          .WORD   65534.
15173  066310  177774          .WORD   65532.
15174  066312  177770          .WORD   65528.
15175  066314  177760          .WORD   65520.
15176  066316  177740          .WORD   65504.
15177  066320  177700          .WORD   65472.
15178  066322  177600          .WORD   65408.
15179  066324  177400          .WORD   65280.
15180  066326  177000          .WORD   65024.
15181  066330  176000          .WORD   64512.
15182  066332  174000          .WORD   63488.
15183  066334  170000          .WORD   61440.
15184  066336  160000          .WORD   57344.
15185  066340  140000          .WORD   49152.
15186  066342  100000          .WORD   32768.
15187  066344  000000          .WORD   0.
15188  066346  000000          .WORD   0.
15189  066350  100000          .WORD   32768.
15190  066352  140000          .WORD   49152.
15191  066354  160000          .WORD   57344.
15192  066356  170000          .WORD   61440.
15193  066360  174000          .WORD   63488.
15194  066362  176000          .WORD   64512.
15195  066364  177000          .WORD   65024.
15196  066366  177400          .WORD   65280.
15197  066370  177600          .WORD   65408.
15198  066372  177700          .WORD   65472.
15199  066374  177740          .WORD   65504.
15200  066376  177760          .WORD   65520.
15201  066400  177770          .WORD   65528.
15202  066402  177774          .WORD   65532.
15203  066404  177776          .WORD   65534.
15204  066406  177777          .WORD   65535.
15205  066410  125252          .WORD   43690.
15206  066412  152525          .WORD   43690./2
15207  066414  125252          .WORD   43690.
15208  066416  177777          .WORD   65535.
15209  066420  177776          .WORD   65534.
15210  066422  177775          .WORD   65533.
15211  066424  177773          .WORD   65531.
15212  066426  177767          .WORD   65527.
15213  066430  177757          .WORD   65519.
15214  066432  177737          .WORD   65503.
15215  066434  177677          .WORD   65471.
15216  066436  177577          .WORD   65407.
15217  066440  177377          .WORD   65279.
15218  066442  176777          .WORD   65023.
15219  066444  175777          .WORD   64511.
15220  066446  173777          .WORD   63487.
15221  066450  167777          .WORD   61439.
15222  066452  157777          .WORD   57343.
15223  066454  137777          .WORD   49151.
15224  066456  077777          .WORD   32767.
```

```
15225 066460 077777                  .WORD   32767.
15226 066462 137777                  .WORD   49151.
15227 066464 157777                  .WORD   57343.
15228 066466 167777                  .WORD   61439.
15229 066470 173777                  .WORD   63487.
15230 066472 175777                  .WORD   64511.
15231 066474 176777                  .WORD   65023.
15232 066476 177377                  .WORD   65279.
15233 066500 177577                  .WORD   65407.
15234 066502 177677                  .WORD   65471.
15235 066504 177737                  .WORD   65503.
15236 066506 177757                  .WORD   65519.
15237 066510 177767                  .WORD   65527.
15238 066512 177773                  .WORD   65531.
15239 066514 177775                  .WORD   65533.
15240 066516 177776                  .WORD   65534.
15241 066520 177777                  .WORD   65535.
15242 066522                 ENRGDT:
```

```
                                        .EVEN

066522  101673  074550  000000  EMT1:   .WORD   EMS300,EMS1,0
066530  101711  101734  101761  EMT2:   .WORD   EMS301,EMS302,EMS303,EMS1,EMS304
066536  074550  101772
066542  104677  104120  104260          .WORD   EMS511,EMS500,EMS501,EMS502,EMS503,0
066550  104305  104354  000000
066556  101711  102001  101734  EMT3:   .WORD   EMS301,EMS306,EMS302
066564  104677  104424  104260          .WORD   EMS511,EMS505,EMS501,EMS502,0
066572  104305  000000
066576  101673  101734  102015  EMT4:   .WORD   EMS300,EMS302,EMS307,EMS2
066604  074621
066606  104677  104305  104260          .WORD   EMS511,EMS502,EMS501,EMS503,0
066614  104354  000000
066620  101711  102056  102073  EMT5:   .WORD   EMS301,EMS310,EMS311
066626  104677  104305  104260          .WORD   EMS511,EMS502,EMS501,EMS503,EMS504
066634  104354  104400
066640  102137  000000                  .WORD   EMS312,0
066644  101711  102001  102073  EMT6:   .WORD   EMS301,EMS306,EMS311
066652  104677  104305  104260          .WORD   EMS511,EMS502,EMS501,EMS503,EMS504,0
066660  104354  104400  000000
066666  101711  102175  101734  EMT7:   .WORD   EMS301,EMS313,EMS302
066674  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,EMS504,EMS503,0
066702  104400  104354  000000
066710  102303  102324  102226  EMT10:  .WORD   EMS316,EMS317,EMS314
066716  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
066724  000000
066726  102303  102324  102255  EMT11:  .WORD   EMS316,EMS317,EMS315
066734  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
066742  000000
066744  102303  102344  102226  EMT12:  .WORD   EMS316,EMS320,EMS314
066752  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
066760  000000
066762  102303  102344  102255  EMT13:  .WORD   EMS316,EMS320,EMS315
066770  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
066776  000000
067000  102303  102364  102226  EMT14:  .WORD   EMS316,EMS321,EMS314
067006  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
067014  000000
067016  102303  102364  102255  EMT15:  .WORD   EMS316,EMS321,EMS315
067024  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
067032  000000
067034  102303  102404  102226  EMT16:  .WORD   EMS316,EMS322,EMS314
067042  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
067050  000000
067052  102303  102404  102255  EMT17:  .WORD   EMS316,EMS322,EMS315
067060  104677  104260  104305          .WORD   EMS511,EMS501,EMS502,0
067066  000000
067070  101711  102443  101161  EMT20:  .WORD   EMS301,EMS324,EMS250
067076  104677  104400                  .WORD   EMS511,EMS504
067102  102137  102474  000000          .WORD   EMS312,EMS326,0
067110  101711  102424  101161  EMT21:  .WORD   EMS301,EMS323,EMS250
067116  104677  104400                  .WORD   EMS511,EMS504
067122  102137  102463  000000          .WORD   EMS312,EMS325,0
067130  101711  102175  101161  EMT22:  .WORD   EMS301,EMS313,EMS250
067136  104677  104400  000000          .WORD   EMS511,EMS504,0
```

```
15299 067144 101711 102443 101217 EMT23: .WORD EMS301,EMS324,EMS251
15300 067152 104677 104260               .WORD EMS511,EMS501
15301 067156 102137 102474 000000        .WORD EMS312,EMS326,0
15302 067164 101711 102424 101217 EMT24: .WORD EMS301,EMS323,EMS251
15303 067172 104677 104260               .WORD EMS511,EMS501
15304 067176 102137 102463 000000        .WORD EMS312,EMS325,0
15305 067204 101711 102175 101217 EMT25: .WORD EMS301,EMS313,EMS251
15306 067212 104677 104260 000000        .WORD EMS511,EMS501,0
15307 067220 101711 102001 101263 EMT26: .WORD EMS301,EMS306,EMS252,EMS253,EMS327,EMS254
15308 067226 101316 102504 101351
15309 067234 104677 104354 104260        .WORD EMS511,EMS503,EMS501,EMS502
15310 067242 104305
15311 067244 102512 102226 000000        .WORD EMS330,EMS314,0
15312 067252 101673 101263        EMT27: .WORD EMS300,EMS252
15313 067256 104677 104260 000000        .WORD EMS511,EMS501,0
15314 067264 101673 101263        EMT30: .WORD EMS300,EMS252
15315 067270 104677 104260 104400        .WORD EMS511,EMS501,EMS504,0
15316 067276 000000
15317 067300 101673 101263        EMT31: .WORD EMS300,EMS252
15318 067304 104677 104260 104354        .WORD EMS511,EMS501,EMS503,0
15319 067312 000000
15320 067314 101711 102443 101263 EMT32: .WORD EMS301,EMS324,EMS252
15321 067322 104677 104260 000000        .WORD EMS511,EMS501,0
15322 067330 101711 102443 101263 EMT33: .WORD EMS301,EMS324,EMS252
15323 067336 104677 104260 104400        .WORD EMS511,EMS501,EMS504,0
15324 067344 000000
15325 067346 101711 102443 101263 EMT34: .WORD EMS301,EMS324,EMS252
15326 067354 104677 104260 104354        .WORD EMS511,EMS501,EMS503,0
15327 067362 000000
15328 067364 101711 102424 101263 EMT35: .WORD EMS301,EMS323,EMS252
15329 067372 104677 104260 000000        .WORD EMS511,EMS501,0
15330 067400 101711 102175 101263 EMT36: .WORD EMS301,EMS313,EMS252
15331 067406 104677 104260 000000        .WORD EMS511,EMS501,0
15332 067414 101711 102443 101412 EMT37: .WORD EMS301,EMS324,EMS255
15333 067422 102137 102474 000000        .WORD EMS312,EMS326,0
15334 067430 101711 102424 101412 EMT40: .WORD EMS301,EMS323,EMS255
15335 067436 104677 104400               .WORD EMS511,EMS504
15336 067442 102137 102463 000000        .WORD EMS312,EMS325,0
15337 067450 101711 102175 101412 EMT41: .WORD EMS301,EMS313,EMS255
15338 067456 104677 104400 000000        .WORD EMS511,EMS504,0
15339 067464 101711 102424 101161 EMT42: .WORD EMS301,EMS323,EMS250,EMS327,EMS255
15340 067472 102504 101412
15341 067476 104677 104400 104260        .WORD EMS511,EMS504,EMS501,EMS503,0
15342 067504 104354 000000
15343 067510 101673 074670        EMT43: .WORD EMS300,EMS3
15344 067514 104677 104260 000000        .WORD EMS511,EMS501,0
15345 067522 102531 074735 101316 EMT44: .WORD EMS331,EMS4,EMS253
15346 067530 104677 104260 000000        .WORD EMS511,EMS501,0
15347
15348 067536 101673 101316        EMT45: .WORD EMS300,EMS253
15349 067542 104677 104260 104354        .WORD EMS511,EMS501,EMS503,0
15350 067550 000000
15351
15352 067552 101673 101316        EMT46: .WORD EMS300,EMS253
15353 067556 104677 104260 104400        .WORD EMS511,EMS501,EMS504,0
15354 067564 000000
```

```

067566  101711  102443  101316  EMT47:  .WORD   EMS301,EMS324,EMS253
067574  104677  104260  104354          .WORD   EMS511,EMS501,EMS503
067602  102137  102474  000000          .WORD   EMS312,EMS326,0

067610  101711  102443  101316  EMT50:  .WORD   EMS301,EMS324,EMS253
067616  104677  104260  104400          .WORD   EMS511,EMS501,EMS504
067624  102137  102474  000000          .WORD   EMS312,EMS326,0

067632  101711  102424  101316  EMT51:  .WORD   EMS301,EMS323,EMS253
067640  104677  104260                  .WORD   EMS511,EMS501
067644  102137  102463  000000          .WORD   EMS312,EMS325,0

067652  101711  102175  101316  EMT52:  .WORD   EMS301,EMS313,EMS253
067660  104677  104260  000000          .WORD   EMS511,EMS501,0

067666  102531  074735  101454  EMT53:  .WORD   EMS331,EMS4,EMS256
067674  104677  104260  000000          .WORD   EMS511,EMS501,0

067702  101711  102443  101454  EMT54:  .WORD   EMS301,EMS324,EMS256
067710  104677  104260                  .WORD   EMS511,EMS501
067714  102137  102474  000000          .WORD   EMS312,EMS326,0

067722  101711  102424  101454  EMT55:  .WORD   EMS301,EMS323,EMS256
067730  104677  104260                  .WORD   EMS511,EMS501
067734  102137  102463  000000          .WORD   EMS312,EMS325,0

067742  101711  102175  101454  EMT56:  .WORD   EMS301,EMS313,EMS256
067750  104677  104260  000000          .WORD   EMS511,EMS501,0

067756  101504  102561          EMT57:  .WORD   EMS257,EMS332
067762  104677  104477  104260          .WORD   EMS511,EMS506,EMS501,0
067770  000000

067772  074766  102577          EMT60:  .WORD   EMS5,EMS333
067776  104677  104534  104260          .WORD   EMS511,EMS507,EMS501,0
070004  000000

070006  101711  102443  101540  EMT61:  .WORD   EMS301,EMS324,EMS260
070014  104677  104260                  .WORD   EMS511,EMS501
070020  102137  102474  000000          .WORD   EMS312,EMS326,0

070026  101711  102424  101540  EMT62:  .WORD   EMS301,EMS323,EMS260
070034  104677  104260                  .WORD   EMS511,EMS501
070040  102137  102463  000000          .WORD   EMS312,EMS325,0

070046  101711  102175  101540  EMT63:  .WORD   EMS301,EMS313,EMS260
070054  104677  104260  000000          .WORD   EMS511,EMS501,0

070062  101673  075307          EMT64:  .WORD   EMS300,EMS12
070066  104677  104260  000000          .WORD   EMS511,EMS501,0

070074  075307  102623  102721  EMT65:  .WORD   EMS12,EMS335,EMS342
070102  104677  104260  000000          .WORD   EMS511,EMS501,0

070110  075307  102646  102721  EMT66:  .WORD   EMS12,EMS336,EMS342
```

```
070116  104677  104260  000000            .WORD   EMS511,EMS501,0

070124  101673  075036          EMT67:    .WORD   EMS300,EMS6
070130  104677  104260                    .WORD   EMS511,EMS501
070134  075102  102577  000000            .WORD   EMS7,EMS333,0

070142  101673  075036  102504  EMT70:    .WORD   EMS300,EMS6,EMS327,EMS7
070150  075102
070152  104677  104260  104400            .WORD   EMS511,EMS501,EMS504,0
070160  000000

070162  075036  102623  102701  EMT71:    .WORD   EMS6,EMS335,EMS340,EMS10,EMS333,EMS342
070170  075153  102577  102721
070176  104677  104260  104305            .WORD   EMS511,EMS501,EMS502,0
070204  000000

070206  075036  102646  102701  EMT72:    .WORD   EMS6,EMS336,EMS340,EMS10,EMS334,EMS342
070214  075153  102613  102721
070222  104677  104260  104305            .WORD   EMS511,EMS501,EMS502,0
070230  000000

070232  101711  101540  101761  EMT73:    .WORD   EMS301,EMS260,EMS303,EMS11
070240  075216
070242  104677  104260  104305            .WORD   EMS511,EMS501,EMS502,0
070250  000000

070252  102753  102767  102721  EMT74:    .WORD   EMS343,EMS344,EMS342,0
070260  000000

070262  101673  075365          EMT75:    .WORD   EMS300,EMS13
070266  104677  104354  000000            .WORD   EMS511,EMS503,0

070274  103044  075365  103016  EMT76:    .WORD   EMS346,EMS13,EMS345
070302  104677  104354  000000            .WORD   EMS511,EMS503,0

070310  102665  075365  103016  EMT77:    .WORD   EMS337,EMS13,EMS345
070316  104677  104354  000000            .WORD   EMS511,EMS503,0

070324  101711  102175  101351  EMT100:   .WORD   EMS301,EMS313,EMS254,EMS347,EMS13
070332  103062  075365
070336  104677  104354  000000            .WORD   EMS511,EMS503,0

070344  103044  075434  102712  EMT101:   .WORD   EMS346,EMS14,EMS341,EMS15
070352  075501
070354  104677  104354  104260            .WORD   EMS511,EMS503,EMS501,0
070362  000000

070364  102665  075434  102712  EMT102:   .WORD   EMS337,EMS14,EMS341,EMS15
070372  075501
070374  104677  104354  104260            .WORD   EMS511,EMS503,EMS501,0
070402  000000

070404  101711  102175  101351  EMT103:   .WORD   EMS301,EMS313,EMS254,EMS347,EMS15
070412  103062  075501
070416  104677  104354                    .WORD   EMS511,EMS503
070422  075434  102561  000000            .WORD   EMS14,EMS332,0
```

```

070430 103044 075640 102712 EMT104: .WORD   EMS346,EMS17,EMS341,EMS16
070436 075557
070440 104677 104354 104260         .WORD   EMS511,EMS503,EMS501,0
070446 000000

070450 102665 075640 102712 EMT105: .WORD   EMS337,EMS17,EMS341,EMS16
070456 075557
070460 104677 104354 104260         .WORD   EMS511,EMS503,EMS501,0
070466 000000

070470 101711 102175 101351 EMT106: .WORD   EMS301,EMS313,EMS254,EMS347,EMS16
070476 103062 075557
070502 104677 104354                .WORD   EMS511,EMS503
070506 075640 102561 000000         .WORD   EMS17,EMS332,0

070514 103044 075701 102712 EMT107: .WORD   EMS346,EMS20,EMS341,EMS21
070522 075745
070524 104677 104354 104260         .WORD   EMS511,EMS503,EMS501,0
070532 000000

070534 075701 103110 075745 EMT110: .WORD   EMS20,EMS351,EMS21,EMS350,EMS22,EMS315
070542 103103 076024 102255
070550 104677 104260 000000         .WORD   EMS511,EMS501,0

070556 075701 102613 103103 EMT111: .WORD   EMS20,EMS334,EMS350,EMS22,EMS333
070564 076024 102577
070570 104677 104260 000000         .WORD   EMS511,EMS501,0

070576 102665 075701 102712 EMT112: .WORD   EMS337,EMS20,EMS341,EMS21
070604 075745
070606 104677 104354 104260         .WORD   EMS511,EMS503,EMS501,0
070614 000000

070616 102665 075701 102712 EMT113: .WORD   EMS337,EMS20,EMS341,EMS21,EMS350,EMS22,EMS334
070624 075745 103103 076024
070632 102613
070634 104677 104260 000000         .WORD   EMS511,EMS501,0

070642 075701 103126 075745 EMT114: .WORD   EMS20,EMS352,EMS21,EMS350,EMS22,EMS333
070650 103103 076024 102577
070656 104677 104260 000000         .WORD   EMS511,EMS501,0

070664 101711 102175 101351 EMT115: .WORD   EMS301,EMS313,EMS254,EMS347,EMS21
070672 103062 075745
070676 104677 104354                .WORD   EMS511,EMS503
070702 075701 102561 000000         .WORD   EMS20,EMS332,0

070710 103044 076071 102712 EMT116: .WORD   EMS346,EMS23,EMS341,EMS24
070716 076147
070720 104677 104354 104260         .WORD   EMS511,EMS503,EMS501,0
070726 000000

070730 102665 076071 102712 EMT117: .WORD   EMS337,EMS23,EMS341,EMS24
070736 076147
070740 104677 104354 104260         .WORD   EMS511,EMS503,EMS501,0
```

```
070746  000000

070750  101711  102175  101351  EMT120: .WORD   EMS301,EMS313,EMS254,EMS347,EMS24
070756  103062  076147
070762  104677  104354                  .WORD   EMS511,EMS503
070766  076071  102561  000000          .WORD   EMS23,EMS332,0

070774  103044  076226  102712  EMT121: .WORD   EMS346,EMS25,EMS341,EMS26
071002  076304
071004  104677  104354  104260          .WORD   EMS511,EMS503,EMS501,0
071012  000000

071014  102665  076226  102712  EMT122: .WORD   EMS337,EMS25,EMS341,EMS26
071022  076304
071024  104677  104354  104260          .WORD   EMS511,EMS503,EMS501,0
071032  000000

071034  101711  102175  101351  EMT123: .WORD   EMS301,EMS313,EMS254,EMS347,EMS26
071042  103062  076304
071046  104677  104354                  .WORD   EMS511,EMS503
071052  076226  102561  000000          .WORD   EMS25,EMS332,0

071060  101673  076363  102015  EMT124: .WORD   EMS300,EMS27,EMS307,EMS2
071066  074621
071070  104677  104354  000000          .WORD   EMS511,EMS503,0

071076  102531  076363  103142  EMT125: .WORD   EMS331,EMS27,EMS353
071104  104677  104354                  .WORD   EMS511,EMS503
071110  076427  102255  000000          .WORD   EMS30,EMS315,0

071116  102665  076363  102712  EMT126: .WORD   EMS337,EMS27,EMS341,EMS30
071124  076427
071126  104677  104354  000000          .WORD   EMS511,EMS503,0

071134  101711  102175  101351  EMT127: .WORD   EMS301,EMS313,EMS254,EMS347,EMS30
071142  103062  076427
071146  104677  104354                  .WORD   EMS511,EMS503
071152  076363  102561  000000          .WORD   EMS27,EMS332,0

071160  076507  103166  101161  EMT130: .WORD   EMS31,EMS354,EMS250
071166  104677  104400  104354          .WORD   EMS511,EMS504,EMS503,0
071174  000000

071176  076560  103166  101161  EMT131: .WORD   EMS32,EMS354,EMS250
071204  104677  104400  104354          .WORD   EMS511,EMS504,EMS503,0
071212  000000

071214  103221  076640  101161  EMT132: .WORD   EMS355,EMS33,EMS250,EMS341,EMS30
071222  102712  076427
071226  104677  104400  000000          .WORD   EMS511,EMS504,0

071234  103221  076677  101161  EMT133: .WORD   EMS355,EMS34,EMS250,EMS341,EMS30
071242  102712  076427
071246  104677  104400  000000          .WORD   EMS511,EMS504,0

071254  102531  074735  101412  EMT134: .WORD   EMS331,EMS4,EMS255
```

```
15579  071262  104677  104400  000000             .WORD   EMS511,EMS504,0
15580
15581  071270  076735  103263  103305  EMT135:   .WORD   EMS35,EMS357,EMS360,EMS15
15582  071276  075501
15583  071300  104677  104260  000000             .WORD   EMS511,EMS501,0
15584
15585  071306  101571  103361          EMT136:   .WORD   EMS261,EMS362
15586  071312  104677  104354  000000             .WORD   EMS511,EMS503,0
15587
15588  071320  101673  076777  102015  EMT137:   .WORD   EMS300,EMS36,EMS307,EMS2
15589  071326  074621
15590  071330  104677  104260  000000             .WORD   EMS511,EMS501,0
15591
15592  071336  103221  077027  101412  EMT140:   .WORD   EMS355,EMS37,EMS255,EMS341,EMS30
15593  071344  102712  076427
15594  071350  104677  104400  000000             .WORD   EMS511,EMS504,0
15595
15596  071356  103044  077067  103016  EMT141:   .WORD   EMS346,EMS40,EMS345
15597  071364  104677  104400  000000             .WORD   EMS511,EMS504,0
15598
15599  071372  102665  077067  102712  EMT142:   .WORD   EMS337,EMS40,EMS341,EMS30
15600  071400  076427
15601  071402  104677  104400  000000             .WORD   EMS511,EMS504,0
15602
15603  071410  103402  102056  077145  EMT143:   .WORD   EMS363,EMS310,EMS41
15604  071416  104677  104260  000000             .WORD   EMS511,EMS501,0
15605
15606  071424  102665  077145  102712  EMT144:   .WORD   EMS337,EMS41,EMS341,EMS252,EMS327,EMS253
15607  071432  101263  102504  101316
15608  071440  104677  104260  000000             .WORD   EMS511,EMS501,0
15609
15610  071446  077145  103166  103417  EMT145:   .WORD   EMS41,EMS354,EMS364,EMS252,EMS365,EMS253
15611  071454  101263  103440  101316
15612  071462  104677  104260  000000             .WORD   EMS511,EMS501,0
15613
15614  071470  101711  102001  076777  EMT146:   .WORD   EMS301,EMS306,EMS36
15615  071476  104677  104260  104354             .WORD   EMS511,EMS501,EMS503,0
15616  071504  000000
15617
15618  071506  103445  077213          EMT147:   .WORD   EMS366,EMS42
15619  071512  104677  104354  000000             .WORD   EMS511,EMS503,0
15620
15621  071520  103470  103142  103440  EMT150:   .WORD   EMS367,EMS353,EMS365,EMS42,EMS354,EMS3
15622  071526  077213  103166  074670
15623  071534  104677  104354  000000             .WORD   EMS511,EMS503,0
15624
15625  071542  102665  076777          EMT151:   .WORD   EMS337,EMS36
15626  071546  104677  104260  000000             .WORD   EMS511,EMS501,0
15627
15628  071554  077275  103166  076777  EMT152:   .WORD   EMS43,EMS354,EMS36
15629  071562  104677  104260  000000             .WORD   EMS511,EMS501,0
15630
15631  071570  103470  103142  103440  EMT153:   .WORD   EMS367,EMS353,EMS365,EMS36,EMS370
15632  071576  076777  103540
15633  071602  104677  104354  000000             .WORD   EMS511,EMS503,0
15634
```

```
071610	103470	103142	103440	EMT154:	.WORD	EMS367,EMS353,EMS365,EMS36,EMS371
071616	076777	103555
071622	104677	104354	000000		.WORD	EMS511,EMS503,0

071630	101673	077346	102015	EMT155:	.WORD	EMS300,EMS44,EMS307,EMS2
071636	074621
071640	104677	104354	000000		.WORD	EMS511,EMS503,0

071646	103470	103142	103440	EMT156:	.WORD	EMS367,EMS353,EMS365,EMS44,EMS354,EMS3
071654	077346	103166	074670
071662	104677	104354	000000		.WORD	EMS511,EMS503,0

071670	101673	077407	102015	EMT157:	.WORD	EMS300,EMS45,EMS307,EMS2
071676	074621
071700	104677	104354	000000		.WORD	EMS511,EMS503,0

071706	103470	103142	103440	EMT160:	.WORD	EMS367,EMS353,EMS365,EMS45,EMS354,EMS3
071714	077407	103166	074670
071722	104677	104354	104260		.WORD	EMS511,EMS503,EMS501
071730	076735	102577	000000		.WORD	EMS35,EMS333,0

071736	101673	077464	102015	EMT161:	.WORD	EMS300,EMS46,EMS307,EMS2
071744	074621
071746	104677	104354	000000		.WORD	EMS511,EMS503,0

071754	102665	077464	103142	EMT162:	.WORD	EMS337,EMS46,EMS353
071762	104677	104354	104260		.WORD	EMS511,EMS503,EMS501,0
071770	000000

071772	076735	102623	102665	EMT163:	.WORD	EMS35,EMS335,EMS337,EMS41,EMS334,EMS372
072000	077145	102613	103604
072006	104677	104260	000000		.WORD	EMS511,EMS501,0

072014	077546	102623	102665	EMT164:	.WORD	EMS47,EMS335,EMS337,EMS41,EMS335,EMS372
072022	077145	102623	103604
072030	104677	104260	000000		.WORD	EMS511,EMS501,0

072036	102665	076735	102504	EMT165:	.WORD	EMS337,EMS35,EMS327,EMS47
072044	077546
072046	104677	104260			.WORD	EMS511,EMS501
072052	077145	102577	103604		.WORD	EMS41,EMS333,EMS372,0
072060	000000

072062	101673	077546	102015	EMT166:	.WORD	EMS300,EMS47,EMS307,EMS2
072070	074621
072072	104677	104260	104354		.WORD	EMS511,EMS501,EMS503,0
072100	000000

072102	077606	102623	102701	EMT167:	.WORD	EMS50,EMS335,EMS340,EMS36,EMS333
072110	076777	102577
072114	104677	104260	104354		.WORD	EMS511,EMS501,EMS503,0
072122	000000

072124	102665	076735		EMT170:	.WORD	EMS337,EMS35
072130	104677	104260	000000		.WORD	EMS511,EMS501,0
```

```
072136  077606  076677  074670  EMT171: .WORD   EMS50,EMS34,EMS3
072144  104677  104260  000000          .WORD   EMS511,EMS501,0

072152  101711  102001  077655  EMT172: .WORD   EMS301,EMS306,EMS51
072160  104677  104260  104400          .WORD   EMS511,EMS501,EMS504,0
072166  000000

072170  103470  103142  103440  EMT173: .WORD   EMS367,EMS353,EMS365,EMS47,EMS354,EMS3
072176  077546  103166  074670
072204  104677  104260  000000          .WORD   EMS511,EMS501,0

072212  101673  101161  102504  EMT174: .WORD   EMS300,EMS250,EMS327,EMS255,EMS327,EMS256
072220  101412  102504  101454
072226  102712  104776                  .WORD   EMS341,EMS600
072232  104677  104260  000000          .WORD   EMS511,EMS501,0

072240  101673  101454  102712  EMT175: .WORD   EMS300,EMS256,EMS341,EMS600
072246  104776
072250  104677  104260  000000          .WORD   EMS511,EMS501,0

072256  101673  101161  102712  EMT176: .WORD   EMS300,EMS250,EMS341,EMS600
072264  104776
072266  104677  104400  000000          .WORD   EMS511,EMS504,0

072274  101673  101412  102712  EMT177: .WORD   EMS300,EMS255,EMS341,EMS600
072302  104776
072304  104677  104400  000000          .WORD   EMS511,EMS504,0

072312  102665  077724  102712  EMT200: .WORD   EMS337,EMS52,EMS341,EMS601
072320  105026
072322  104677  104260  000000          .WORD   EMS511,EMS501,0

072330  103044  077724  102712  EMT201: .WORD   EMS346,EMS52,EMS341,EMS602
072336  105046
072340  104677  104260  000000          .WORD   EMS511,EMS501,0

072346  103044  077655  102712  EMT202: .WORD   EMS346,EMS51,EMS341,EMS602
072354  105046
072356  104677  104260  000000          .WORD   EMS511,EMS501,0

072364  103044  077724  103016  EMT203: .WORD   EMS346,EMS52,EMS345,EMS373,EMS255
072372  103636  101412
072376  104677  104400  104260          .WORD   EMS511,EMS504,EMS501,0
072404  000000

072406  103044  077724  102712  EMT204: .WORD   EMS346,EMS52,EMS341,EMS27
072414  076363
072416  104677  104400  104260          .WORD   EMS511,EMS504,EMS501,0
072424  000000

072426  077765  103166  074670  EMT205: .WORD   EMS53,EMS354,EMS3
072434  104677  104354  104305          .WORD   EMS511,EMS503,EMS502,EMS510,0
072442  104624  000000

072446  102665  100026  103642  EMT206: .WORD   EMS337,EMS54,EMS374,EMS250,EMS327,EMS255
072454  101161  102504  101412
```

```
072462  102504  074670                    .WORD   EMS327,EMS3
072466  104677  104400  104260            .WORD   EMS511,EMS504,EMS501,0
072474  000000

072476  100026  103166  074670  EMT207:   .WORD   EMS54,EMS354,EMS3
072504  104677  104260  000000            .WORD   EMS511,EMS501,0

072512  100026  103166  103417  EMT210:   .WORD   EMS54,EMS354,EMS364,EMS250
072520  101161
072522  104677  104400  000000            .WORD   EMS511,EMS504,0

072530  100026  103166  103417  EMT211:   .WORD   EMS54,EMS354,EMS364,EMS255
072536  101412
072540  104677  104400  000000            .WORD   EMS511,EMS504,0

072546  102665  100103          EMT212:   .WORD   EMS337,EMS55
072552  104677  104400  000000            .WORD   EMS511,EMS504,0

072560  100161  102613  103016  EMT213:   .WORD   EMS56,EMS334,EMS345,EMS373,EMS262,EMS327,EMS251
072566  103636  101630  102504
072574  101217
072576  104677  104260                    .WORD   EMS511,EMS501
072602  100161  102623  000000            .WORD   EMS56,EMS335,0

072610  102665  100161          EMT214:   .WORD   EMS337,EMS56
072614  104677  104260  000000            .WORD   EMS511,EMS501,0

072622  100254  102577  103103  EMT215:   .WORD   EMS57,EMS333,EMS350,EMS60,EMS334
072630  100340  102613
072634  104677  104260  000000            .WORD   EMS511,EMS501,0

072642  103402  102001  074766  EMT216:   .WORD   EMS363,EMS306,EMS5
072650  104677  104260  000000            .WORD   EMS511,EMS501,0

072656  103402  102001  100412  EMT217:   .WORD   EMS363,EMS306,EMS61
072664  104677  104260  000000            .WORD   EMS511,EMS501,0

072672  100465  102577  103673  EMT220:   .WORD   EMS62,EMS333,EMS375,EMS251
072700  101217
072702  104677  104260  000000            .WORD   EMS511,EMS501,0

072710  100465  102577  103707  EMT221:   .WORD   EMS62,EMS333,EMS376,EMS262
072716  101630
072720  104677  104260  000000            .WORD   EMS511,EMS501,0

072726  100465  102577  103707  EMT222:   .WORD   EMS62,EMS333,EMS376,EMS250
072734  101161
072736  104677  104260  000000            .WORD   EMS511,EMS501,0

072744  103044  100465  102712  EMT223:   .WORD   EMS346,EMS62,EMS341,EMS603
072752  105107
072754  104677  104260  000000            .WORD   EMS511,EMS501,0

072762  103044  100546  103707  EMT224:   .WORD   EMS346,EMS63,EMS376,EMS262
072770  101630
072772  104677  104260  104354            .WORD   EMS511,EMS501,EMS503,0
```

```
073000  000000

073002  100546  102577  103103  EMT225: .WORD   EMS63,EMS333,EMS350,EMS363,EMS310,EMS262
073010  103402  102056  101630
073016  104677  104260  000000          .WORD   EMS511,EMS501,0

073024  100546  103263  076777  EMT226: .WORD   EMS63,EMS357,EMS36,EMS372
073032  103604
073034  104677  104260  000000          .WORD   EMS511,EMS501,0

073042  100546  103243  103305  EMT227: .WORD   EMS63,EMS356,EMS360,EMS15
073050  075501
073052  104677  104260  000000          .WORD   EMS511,EMS501,0

073060  100546  103243  103333  EMT230: .WORD   EMS63,EMS356,EMS361,EMS15
073066  075501
073070  104677  104260  000000          .WORD   EMS511,EMS501,0

073076  100546  103243  077145  EMT231: .WORD   EMS63,EMS356,EMS41
073104  104677  104260  000000          .WORD   EMS511,EMS501,0

073112  077145  103723  102721  EMT232: .WORD   EMS41,EMS377,EMS342,EMS365,EMS63,EMS332
073120  103440  100546  102561
073126  104677  104260  000000          .WORD   EMS511,EMS501,0

073134  103470  103142  103440  EMT233: .WORD   EMS367,EMS353,EMS365,EMS63,EMS401
073142  100546  103751
073146  104677  104354  104260          .WORD   EMS511,EMS503,EMS501,0
073154  000000

073156  075557  103723  103604  EMT234: .WORD   EMS16,EMS377,EMS372,EMS365,EMS64,EMS354,EMS3
073164  103440  100606  103166
073172  074670
073174  104677  104354  104260          .WORD   EMS511,EMS503,EMS501,0
073202  000000

073204  101673  100656  102015  EMT235: .WORD   EMS300,EMS65,EMS307,EMS2
073212  074621
073214  104677  104305  104354          .WORD   EMS511,EMS502,EMS503,0
073222  000000

073224  100161  103723  103707  EMT236: .WORD   EMS56,EMS377,EMS376,EMS252,EMS372,EMS350
073232  101263  103604  103103
073240  100656  103751                  .WORD   EMS65,EMS401
073244  104677  104305  104354          .WORD   EMS511,EMS502,EMS503,EMS501,0
073252  104260  000000

073256  101673  100717  102015  EMT237: .WORD   EMS300,EMS66,EMS307,EMS2
073264  074621
```

F09


```
073266  104677  104260  104354           .WORD   EMS511,EMS501,EMS503,0
073274  000000

073276  075501  103734  103604  EMT240:  .WORD   EMS15,EMS400,EMS372,EMS350,EMS66,EMS401
073304  103103  100717  103751
073312  104677  104354  104260           .WORD   EMS511,EMS503,EMS501,0
073320  000000

073322  100717  102646  102701  EMT241:  .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS604
073330  075501  104065  104055
073336  105130
073340  104677  104354  000000           .WORD   EMS511,EMS503,0

073346  103775  105130  103766  EMT242:  .WORD   EMS403,EMS604,EMS402,EMS21,EMS377
073354  075745  103723
073360  104677  104354                   .WORD   EMS511,EMS503
073364  076777  102623  000000           .WORD   EMS36,EMS335,0

073372  100717  102646  102701  EMT243:  .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS604
073400  076304  104036  104055
073406  105130
073410  104677  104354  000000           .WORD   EMS511,EMS503,0

073416  100546  103751  104055  EMT244:  .WORD   EMS63,EMS401,EMS405,EMS604
073424  105130
073426  104677  104354  000000           .WORD   EMS511,EMS503,0

073434  076777  103540  104055  EMT245:  .WORD   EMS36,EMS370,EMS405,EMS604
073442  105130
073444  104677  104354  000000           .WORD   EMS511,EMS503,0

073452  103775  105130  103766  EMT246:  .WORD   EMS403,EMS604,EMS402,EMS24,EMS377
073460  076147  103723
073464  104677  104354                   .WORD   EMS511,EMS503
073470  076777  102623  000000           .WORD   EMS36,EMS335,0

073476  101001  102561  104055  EMT247:  .WORD   EMS67,EMS332,EMS405,EMS604
073504  105130
073506  104677  104354  000000           .WORD   EMS511,EMS503,0

073514  100717  102646  102701  EMT250:  .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS605
073522  075501  104065  104055
073530  105155
073532  104677  104354  000000           .WORD   EMS511,EMS503,0

073540  103775  105155  103766  EMT251:  .WORD   EMS403,EMS605,EMS402,EMS21,EMS377
073546  075745  103723
073552  104677  104354                   .WORD   EMS511,EMS503
073556  076777  102623  000000           .WORD   EMS36,EMS335,0

073564  100717  102646  102701  EMT252:  .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS605
073572  076304  104036  104055
073600  105155
073602  104677  104354  000000           .WORD   EMS511,EMS503,0

073610  100546  103751  104055  EMT253:  .WORD   EMS63,EMS401,EMS405,EMS605
```

```
073616  105155
073620  104677  104354  000000            .WORD   EMS511,EMS503,0

073626  076777  103540  104055  EMT254:  .WORD   EMS36,EMS370,EMS405,EMS605
073634  105155
073636  104677  104354  000000            .WORD   EMS511,EMS503,0

073644  103775  105155  103766  EMT255:  .WORD   EMS403,EMS605,EMS402,EMS24,EMS377
073652  076147  103723
073656  104677  104354                    .WORD   EMS511,EMS503
073662  076777  102623  000000            .WORD   EMS36,EMS335,0

073670  101001  102561  104055  EMT256:  .WORD   EMS67,EMS332,EMS405,EMS605
073676  105155
073700  104677  104354  000000            .WORD   EMS511,EMS503,0

073706  100717  102646  102701  EMT257:  .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS606
073714  075501  104065  104055
073722  105173
073724  104677  104354  000000            .WORD   EMS511,EMS503,0

073732  103775  105173  103766  EMT260:  .WORD   EMS403,EMS606,EMS402,EMS21,EMS377
073740  075745  103723
073744  104677  104354                    .WORD   EMS511,EMS503
073750  076777  102623  000000            .WORD   EMS36,EMS335,0

073756  100717  102646  102701  EMT261:  .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS606
073764  076304  104036  104055
073772  105173
073774  104677  104354  000000            .WORD   EMS511,EMS503,0

074002  100546  103751  104055  EMT262:  .WORD   EMS63,EMS401,EMS405,EMS606
074010  105173
074012  104677  104354  000000            .WORD   EMS511,EMS503,0

074020  076777  103540  104055  EMT263:  .WORD   EMS36,EMS370,EMS405,EMS606
074026  105173
074030  104677  104354  000000            .WORD   EMS511,EMS503,0

074036  103775  105173  103766  EMT264:  .WORD   EMS403,EMS606,EMS402,EMS24,EMS377
074044  076147  103723
074050  104677  104354                    .WORD   EMS511,EMS503
074054  076777  102623  000000            .WORD   EMS36,EMS335,0

074062  101113  103751  104055  EMT265:  .WORD   EMS70,EMS401,EMS405,EMS606
074070  105173
074072  104677  104354  000000            .WORD   EMS511,EMS503,0

074100  101001  102561  104055  EMT266:  .WORD   EMS67,EMS332,EMS405,EMS606
074106  105173
074110  104677  104354  000000            .WORD   EMS511,EMS503,0

074116  100717  103243  104100  EMT267:  .WORD   EMS66,EMS356,EMS407
074124  104677  104354  000000            .WORD   EMS511,EMS503,0

074132  100717  102646  102701  EMT270:  .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS607
```

```
074140  075501  104065  104055
074146  105213
074150  104677  104354  000000          .WORD   EMS511,EMS503,0

074156  103775  105213  103766  EMT271: .WORD   EMS403,EMS607,EMS402,EMS21,EMS377
074164  075745  103723
074170  104677  104354                  .WORD   EMS511,EMS503
074174  076363  102646  000000          .WORD   EMS27,EMS336,0

074202  076363  103540  104055  EMT272: .WORD   EMS27,EMS370,EMS405,EMS607
074210  105213
074212  104677  104354  000000          .WORD   EMS511,EMS503,0

074220  076363  103555  104055  EMT273: .WORD   EMS27,EMS371,EMS405,EMS607
074226  105213
074230  104677  104354  000000          .WORD   EMS511,EMS503,0

074236  076777  104036  104055  EMT274: .WORD   EMS36,EMS404,EMS405,EMS607
074244  105213
074246  104677  104354  000000          .WORD   EMS511,EMS503,0

074254  077765  103751  104055  EMT275: .WORD   EMS53,EMS401,EMS405,EMS607
074262  105213
074264  104677  104354  104305          .WORD   EMS511,EMS503,EMS502,0
074272  000000

074274  101001  102561  104055  EMT276: .WORD   EMS67,EMS332,EMS405,EMS607
074302  105213
074304  104677  104354  000000          .WORD   EMS511,EMS503,0

074312  100717  102646  102701  EMT277: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS607
074320  076304  104036  104055
074326  105213
074330  104677  104354  000000          .WORD   EMS511,EMS503,0

074336  103775  105213  103766  EMT300: .WORD   EMS403,EMS607,EMS402,EMS24,EMS377
074344  076147  103723
074350  104677  104354                  .WORD   EMS511,EMS503
074354  076363  102646  000000          .WORD   EMS27,EMS336,0

074362  101113  103751  104055  EMT301: .WORD   EMS70,EMS401,EMS405,EMS607
074370  105213
074372  104677  104354  000000          .WORD   EMS511,EMS503,0

074400  105231  000000          EHT1:   .WORD   EH1,0
074404  105310  000000          EHT2:   .WORD   EH2,0
074410  105326  000000          EHT5:   .WORD   EH5,0
074414  105365  000000          EHT7:   .WORD   EH7,0
074420  105404  000000          EHT57:  .WORD   EH57,0
074424  105503  000000          EHT65:  .WORD   EH65,0
074430  105560  000000          EHT71:  .WORD   EH71,0
074434  105317  000000          EHT74:  .WORD   EH3,0
074440  105636  000000          EHT115: .WORD   EH115,0
074444  105734  000000          EHT130: .WORD   EH130,0
074450  106053  000000          EHT132: .WORD   EH132,0
074454  106152  000000          EHT145: .WORD   EH145,0
```

```
074460  106270  000000                  EHT150: .WORD   EH150,0
074464  106367  000000                  EHT213: .WORD   EH213,0
074470  106465  000000                  EHT220: .WORD   EH220,0

074474  106524                          EDT1:   .WORD   ED1
074476  106534                          EDT2:   .WORD   ED2
074500  106540                          EDT5:   .WORD   ED5
074502  106546                          EDT57:  .WORD   ED57
074504  106560                          EDT65:  .WORD   ED65
074506  106570                          EDT71:  .WORD   ED71
074510  106534                          EDT74:  .WORD   ED2
074512  106600                          EDT115: .WORD   ED115
074514  106612                          EDT130: .WORD   ED130
074516  106600                          EDT132: .WORD   ED115
074520  106626                          EDT220: .WORD   ED220

074522  106632                          EFT1:   .WORD   EF1
074524  106635                          EFT2:   .WORD   EF2
074526  106636                          EFT5:   .WORD   EF5
074530  106640                          EFT57:  .WORD   EF57
074532  106632                          EFT65:  .WORD   EF1
074534  106632                          EFT71:  .WORD   EF1
074536  106635                          EFT74:  .WORD   EF2
074540  106640                          EFT115: .WORD   EF57
074542  106644                          EFT130: .WORD   EF130
074544  106640                          EFT132: .WORD   EF57
074546  106636                          EFT220: .WORD   EF5

074550  047516  042516  044530          EMS1:   .ASCIZ  @NONEXISTENT DEVICE "NED" (RMCS2,BIT 12) @
074556  052123  047105  020124
074564  042504  044526  042503
074572  021040  042516  021104
074600  024040  046522  051503
074606  026062  044502  020124
074614  031061  020051     000
074621     103  047117  051124          EMS2:   .ASCIZ  @CONTROLLER CLEAR "CLR" (RMCS2,BIT 05) @
074626  046117  042514  020122
074634  046103  040505  020122
074642  041442  051114  020042
074650  051050  041515  031123
074656  041054  052111  030040
074664  024465  000040
074670  052506  041516  044524          EMS3:   .ASCIZ  @FUNCTION CODE (RMCS1, BITS 01 - 05) @
074676  047117  041440  042117
074704  020105  051050  041515
074712  030523  020054  044502
074720  051524  030040  020061
074726  020055  032460  020051
074734     000
074735     125  052516  042523          EMS4:   .ASCIZ  @UNUSED BIT POSITIONS OF @
074742  020104  044502  020124
074750  047520  044523  044524
074756  047117  020123  043117
074764  000040
074766  042504  044526  042503          EMS5:   .ASCIZ  @DEVICE AVAILABLE "DVA" (RMCS1, BIT 11) @
074774  040440  040526  046111
```

```
16076  075002  041101  042514  021040
16077  075010  053104  021101  024040
16078  075016  046522  051503  026061
16079  075024  041040  052111  030440
16080  075032  024461  000040
16081  075036  040520  052122  054511  EMS6:   .ASCIZ  @PARTIY ERROR "PAR" (RMER1, BIT 03) @
16082  075044  042440  051122  051117
16083  075052  021040  040520  021122
16084  075060  024040  046522  051105
16085  075066  026061  041040  052111
16086  075074  030040  024463  000040
16087  075102  040504  040524  050040  EMS7:   .ASCIZ  @DATA PARITY ERROR "DPE" (RMER2, BIT 03) @
16088  075110  051101  052111  020131
16089  075116  051105  047522  020122
16090  075124  042042  042520  020042
16091  075132  051050  042515  031122
16092  075140  020054  044502  020124
16093  075146  031460  020051     000
16094  075153     120  051101  052111  EMS10:  .ASCIZ  @PARITY TEST "PAT" (RMCS2, BIT 04) @
16095  075160  020131  042524  052123
16096  075166  021040  040520  021124
16097  075174  024040  046522  051503
16098  075202  026062  041040  052111
16099  075210  030040  024464  000040
16100  075216  040515  051523  052502  EMS11:  .ASCII  @MASSBUS CONTROL BUS PARITY ERROR "MCPE" @
16101  075224  020123  047503  052116
16102  075232  047522  020114  052502
16103  075240  020123  040520  044522
16104  075246  054524  042440  051122
16105  075254  051117  021040  041515
16106  075262  042520  020042
16107  075266  051050  041515  030523          .ASCIZ  @(RMCS1, BIT 13) @
16108  075274  020054  044502  020124
16109  075302  031461  020051     000
16110  075307     111  046114  043505  EMS12:  .ASCIZ  @ILLEGAL REGISTER ERROR "ILR" (RMER1, BIT 01) @
16111  075314  046101  051040  043505
16112  075322  051511  042524  020122
16113  075330  051105  047522  020122
16114  075336  044442  051114  020042
16115  075344  051050  042515  030522
16116  075352  020054  044502  020124
16117  075360  030460  020051     000
16118  075365     104  040511  047107  EMS13:  .ASCIZ  @DIAGNOSTIC MODE "DMD" (RMMR1, BIT 00) @
16119  075372  051517  044524  020103
16120  075400  047515  042504  021040
16121  075406  046504  021104  024040
16122  075414  046522  051115  026061
16123  075422  041040  052111  030040
16124  075430  024460  000040
16125  075434  042515  044504  046525  EMS14:  .ASCIZ  @MEDIUM ON LINE "MOL" (RMDS, BIT 12) @
16126  075442  047440  020116  044514
16127  075450  042516  021040  047515
16128  075456  021114  024040  046522
16129  075464  051504  020054  044502
16130  075472  020124  031061  020051
16131  075500     000
```

```
075501     115  044501  052116   EMS15:  .ASCIZ  @MAINTENANCE UNIT READY "MUR" (RMMR1, BIT 09) @
075506  047105  047101  042503
075514  052440  044516  020124
075522  042522  042101  020131
075530  046442  051125  020042
075536  051050  046515  030522
075544  020054  044502  020124
075552  034460  020051     000
075557     115  044501  052116   EMS16:  .ASCIZ  @MAINTENANCE WRITE PROTECT "MWP" (RMMR1, BIT 03) @
075564  047105  047101  042503
075572  053440  044522  042524
075600  050040  047522  042524
075606  052103  021040  053515
075614  021120  024040  046522
075622  051115  026061  041040
075630  052111  030040  024463
075636  000040
075640  051127  052111  020105   EMS17:  .ASCIZ  @WRITE LOCK "WRL" (RMDS, BIT 11) @
075646  047514  045503  021040
075654  051127  021114  024040
075662  046522  051504  020054
075670  044502  020124  030461
075676  020051     000
075701     104  053105  041511   EMS20:  .ASCIZ  @DEVICE CHECK "DVC" (RMER2, BIT 07) @
075706  020105  044103  041505
075714  020113  042042  041526
075722  020042  051050  042515
075730  031122  020054  044502
075736  020124  033460  020051
075744     000
075745     115  044501  052116   EMS21:  .ASCIZ  @MAINTENANCE DRIVE FAULT "MDF" (RMMR1, BIT 06) @
075752  047105  047101  042503
075760  042040  044522  042526
075766  043040  052501  052114
075774  021040  042115  021106
076002  024040  046522  051115
076010  026061  041040  052111
076016  030040  024466  000040
076024  047125  040523  042506   EMS22:  .ASCIZ  @UNSAFE STATUS "UNS" (RMER1, BIT 14) @
076032  051440  040524  052524
076040  020123  052442  051516
076046  020042  051050  042515
076054  030522  020054  044502
076062  020124  032061  020051
076070     000
076071     123  042505  020113   EMS23:  .ASCIZ  @SEEK INCOMPLETE STATUS "SKI" (RMER2, BIT 14) @
076076  047111  047503  050115
076104  042514  042524  051440
076112  040524  052524  020123
076120  051442  044513  020042
076126  051050  042515  031122
076134  020054  044502  020124
076142  032061  020051     000
076147     115  044501  052116   EMS24:  .ASCIZ  @MAINTENANCE SEEK ERROR "MSER" (RMMR1, BIT 07) @
076154  047105  047101  042503
076162  051440  042505  020113
```

```
076170  051105  047522  020122
076176  046442  042523  021122
076204  024040  046522  051115
076212  026061  041040  052111
076220  030040  024467  000040
076226  047520  044523  044524   EMS25:  .ASCIZ  @POSITIONING IN PROGRESS "PIP" (RMDS, BIT 13) @
076234  047117  047111  020107
076242  047111  050040  047522
076250  051107  051505  020123
076256  050042  050111  020042
076264  051050  042115  026123
076272  041040  052111  030440
076300  024463  000040
076304  040515  047111  042524   EMS26:  .ASCIZ  @MAINTENANCE ON CYLINDER "MOC" (RMMR1, BIT 08) @
076312  040516  041516  020105
076320  04711[illegible]  041440  046131
076326  047111  042504  020122
076334  046442  041517  020042
076342  051050  046515  030522
076350  020054  044502  020124
076356  034060  020051     000
076363     105  042116  047440   EMS27:  .ASCIZ  @END OF BLOCK "EBL" (RMMR1, BIT 13) @
076370  020106  046102  041517
076376  020113  042442  046102
076404  020042  051050  046515
076412  030522  020054  044502
076420  020124  031461  020051
076426     000
076427     104  040511  047107   EMS30:  .ASCIZ  @DIAGNOSTIC END OF BLOCK "DEBL" (RMMR1, BIT 13) @
076434  051517  044524  020103
076442  047105  020104  043117
076450  041040  047514  045503
076456  021040  042504  046102
076464  020042  051050  046515
076472  030522  020054  044502
076500  020124  031461  020051
076506     000
076507     114  051501  020124   EMS31:  .ASCIZ  @LAST SECTOR STATUS "LS" (RMMR1, BIT 02) @
076514  042523  052103  051117
076522  051440  040524  052524
076530  020123  046042  021123
076536  024040  046522  051115
076544  026061  041040  052111
076552  030040  024462  000040
076560  040514  052123  051440   EMS32:  .ASCIZ  @LAST SECTOR/TRACK STATUS "LST" (RMMR1, BIT 01) @
076566  041505  047524  027522
076574  051124  041501  020113
076602  052123  052101  051525
076610  021040  051514  021124
076616  024040  046522  051115
076624  026061  041040  052111
076632  030040  024461  000040
076640  042523  052103  051117   EMS33:  .ASCIZ  @SECTOR ADDRESS, BITS 00-04 OF @
076646  040440  042104  042522
076654  051523  020054  044502
076662  051524  030040  026460
```

```
076670  032060  047440  020106
076676     000
076677     124  040522  045503  EMS34:  .ASCIZ  @TRACK ADDRESS, BITS 08-10 OF @
076704  040440  042104  042522
076712  051523  020054  044502
076720  051524  030040  026470
076726  030061  047440  020106
076734     000
076735     126  046117  046525  EMS35:  .ASCIZ  @VOLUME VALID "VV" (RMDS, BIT 06) @
076742  020105  040526  044514
076750  020104  053042  021126
076756  024040  046522  051504
076764  020054  044502  020124
076772  033060  020051     000
076777     107  020117  044502  EMS36:  .ASCIZ  @GO BIT (RMCS1, BIT 00) @
077004  020124  051050  041515
077012  030523  020054  044502
077020  020124  030060  020051
077026     000
077027     103  046131  047111  EMS37:  .ASCIZ  @CYLINDER ADDRESS, BIT 00-09 OF @
077034  042504  020122  042101
077042  051104  051505  026123
077050  041040  052111  030040
077056  026460  034460  047440
077064  020106     000
077067     114  051501  020124  EMS40:  .ASCIZ  @LAST BLOCK TAKEN STATUS "LBT" (RMDS, BIT 10) @
077074  046102  041517  020113
077102  040524  042513  020116
077110  052123  052101  051525
077116  021040  041114  021124
077124  024040  046522  051504
077132  020054  044502  020124
077140  030061  020051     000
077145     103  046517  047520  EMS41:  .ASCIZ  @COMPOSITE ERROR "ERR" (RMDS, BIT 14) @
077152  044523  042524  042440
077160  051122  051117  021040
077166  051105  021122  024040
077174  046522  051504  020054
077202  044502  020124  032061
077210  020051     000
077213     103  046517  040515  EMS42:  .ASCIZ  @COMMAND SEQUENCER TEST BIT "TST" (RMMR2, BIT 12) @
077220  042116  051440  050505
077226  042525  041516  051105
077234  052040  051505  020124
077242  044502  020124  052042
077250  052123  020042  051050
077256  046515  031122  020054
077264  044502  020124  031061
077272  020051     000
```

```
077275     104 044522 042526  EMS43:  .ASCIZ  @DRIVE READY STATUS "DRY" (RMDS, BIT 07) @
077302  051040 040505 054504
077310  051440 040524 052524
077316  020123 042042 054522
077324  020042 051050 042115
077332  026123 041040 052111
077340  030040 024467 000040
077346  047503 052116 047111  EMS44:  .ASCIZ  @CONTINUE "CONT" (RMMR1, BIT 06) @
077354  042525 021040 047503
077362  052116 020042 051050
077370  046515 030522 020054
077376  044502 020124 033060
077404  020051    000
077407     111 053116 046101  EMS45:  .ASCIZ  @INVALID COMMAND ERROR "IVC" (RMER2, BIT 12) @
077414  042111 041440 046517
077422  040515 042116 042440
077430  051122 051117 021040
077436  053111 021103 024040
077444  046522 051105 026062
077452  041040 052111 030440
077460  024462 000040
077464  047514 051523 047440  EMS46:  .ASCIZ  @LOSS OF SYSTEM CLOCK ERROR "LSC" (RMER2, BIT 11) @
077472  020106 054523 052123
077500  046505 041440 047514
077506  045503 042440 051122
077514  051117 021040 051514
077522  021103 024040 046522
077530  051105 026062 041040
077536  052111 030440 024461
077544  000040
077546  041517 052503 044520  EMS47:  .ASCIZ  @OCCUPIED "OCC" (RMMR1, BIT 15) @
077554  042105 021040 041517
077562  021103 024040 046522
077570  051115 026061 041040
077576  052111 030440 024465
077604  000040
077606  046111 042514 040507  EMS50:  .ASCIZ  @ILLEGAL FUNCTION "ILF" (RMER1, BIT 0) @
077614  020114 052506 041516
077622  044524 047117 021040
077630  046111 021106 024040
077636  046522 051105 026061
077644  041040 052111 030040
077652  020051    000
077655     117 043106 042523  EMS51:  .ASCIZ  @OFFSET DIRECTION "OFD" (RMOF, BIT 07) @
077662  020124 044504 042522
077670  052103 047511 020116
077676  047442 042106 020042
077704  051050 047515 026106
077712  041040 052111 030040
077720  024467 000040
077724  043117 051506 052105  EMS52:  .ASCIZ  @OFFSET MODE "OM" (RMDS, BIT 00) @
077732  046440 042117 020105
077740  047442 021115 024040
077746  046522 051504 020054
077754  044502 020124 030060
077762  020051    000
```

```
16349 077765    122 047125 040440 EMS53:  .ASCIZ  @RUN AND GO "RG" (RMMR1, BIT 14) @
16350 077772 042116 043440 020117
16351 100000 051042 021107 024040
16352 100006 046522 051115 026061
16353 100014 041040 052111 030440
16354 100022 024464 000040
16355 100026 047111 040526 044514 EMS54:  .ASCIZ  @INVALID ADDRESS ERROR "IAE" (RMER1, BIT 10) @
16356 100034 020104 042101 051104
16357 100042 051505 020123 051105
16358 100050 047522 020122 044442
16359 100056 042501 020042 051050
16360 100064 042515 030522 020054
16361 100072 044502 020124 030061
16362 100100 020051    000
16363 100103    101 042104 042522 EMS55:  .ASCIZ  @ADDRESS OVERFLOW ERROR "AOE" (RMER1, BIT 09) @
16364 100110 051523 047440 042526
16365 100116 043122 047514 020127
16366 100124 051105 047522 020122
16367 100132 040442 042517 020042
16368 100140 051050 042515 030522
16369 100146 020054 044502 020124
16370 100154 034460 020051    000
16371 100161    122 043505 051511 EMS56:  .ASCII  @REGISTER MODIFICATION REFUSED ERROR @
16372 100166 042524 020122 047515
16373 100174 044504 044506 040503
16374 100202 044524 047117 051040
16375 100210 043105 051525 042105
16376 100216 042440 051122 051117
16377 100224    040
16378 100225    042 046522 021122         .ASCIZ  @"RMR" (RMER1, BIT 02) @
16379 100232 024040 046522 051105
16380 100240 026061 041040 052111
16381 100246 030040 024462 000040
16382 100254 051104 053111 020105 EMS57:  .ASCIZ  @DRIVE REQUEST REQUIRED STATUS "DRQ" (RMDT, BIT 11) @
16383 100262 042522 052521 051505
16384 100270 020124 042522 052521
16385 100276 051111 042105 051440
16386 100304 040524 052524 020123
16387 100312 042042 050522 020042
16388 100320 051050 042115 026124
16389 100326 041040 052111 030440
16390 100334 024461 000040
16391 100340 051120 043517 040522 EMS60:  .ASCIZ  @PROGRAMMABLE STATUS "PGM" (RMDS, BIT 09) @
16392 100346 046515 041101 042514
16393 100354 051440 040524 052524
16394 100362 020123 050042 046507
16395 100370 020042 051050 042115
16396 100376 026123 041040 052111
16397 100404 030040 024471 000040
16398 100412 051104 053111 020105 EMS61:  .ASCIZ  @DRIVE PRESENT STATUS "DPR" (RMDS, BIT 08) @
16399 100420 051120 051505 047105
16400 100426 020124 052123 052101
16401 100434 051525 021040 050104
16402 100442 021122 024040 046522
16403 100450 051504 020054 044502
16404 100456 020124 034060 020051
```

```
100464     000
100465     120  051117  020124  EMS62:  .ASCIZ  @PORT REQUEST FLOP "RQA,RQB" (RMMR2, BITS 15,14) @
100472  042522  052521  051505
100500  020124  046106  050117
100506  021040  050522  026101
100514  050522  021102  024040
100522  046522  051115  026062
100530  041040  052111  020123
100536  032461  030454  024464
100544  000040
100546  052101  042524  052116  EMS63:  .ASCIZ  @ATTENTION "ATA" (RMDS, BIT 15) @
100554  047511  020116  040442
100562  040524  020042  051050
100570  042115  026123  041040
100576  052111  030440  024465
100604  000040
100606  051127  052111  020105  EMS64:  .ASCIZ  @WRITE LOCK ERROR "WLE" (RMER1, BIT 11) @
100614  047514  045503  042440
100622  051122  051117  021040
100630  046127  021105  024040
100636  046522  051105  026061
100644  041040  052111  030440
100652  024461  000040
100656  054105  042503  052120  EMS65:  .ASCIZ  @EXCEPTION "REX" (RMMR1, BIT 12) @
100664  047511  020116  051042
100672  054105  020042  051050
100700  046515  030522  020054
100706  044502  020124  031061
100714  020051     000
100717     117  042520  040522  EMS66:  .ASCIZ  @OPERATION INCOMPLETE ERROR "OPI" (RMER1, BIT 13) @
100724  044524  047117  044440
100732  041516  046517  046120
100740  052105  020105  051105
100746  047522  020122  047442
100754  044520  020042  051050
100762  042515  030522  020054
100770  044502  020124  031461
100776  020051     000
101001     124  043501  041040  EMS67:  .ASCIZ  @TAG BUS (RMMR2, BITS 00-09) OR TAG CONTROL @
101006  051525  024040  046522
101014  051115  026062  041040
101022  052111  020123  030060
101030  030055  024471  047440
101036  020122  040524  020107
101044  047503  052116  047522
101052  020114     000
101055     114  047111  051505          .ASCIZ  @LINES (RMMR2, BITS 10,11,13) @
101062  024040  046522  051115
101070  026062  041040  052111
101076  020123  030061  030454
101104  026061  031461  020051
101112     000
101113     123  040505  041522  EMS70:  .ASCIZ  @SEARCH ENABLE "ESRC" (RMMR1, BIT 11) @
101120  020110  047105  041101
101126  042514  021040  051505
101134  041522  020042  051050
```

```
101142  046515  030522  020054
101150  044502  020124  030461
101156  020051     000

101161     104  051511  020113  EMS250: .ASCIZ  @DISK ADDRESS REGISTER (RMDA) @
101166  042101  051104  051505
101174  020123  042522  044507
101202  052123  051105  024040
101210  046522  040504  020051
101216     000
101217     103  047117  051124  EMS251: .ASCIZ  @CONTROL STATUS REGISTER #1 (RMCS1) @
101224  046117  051440  040524
101232  052524  020123  042522
101240  044507  052123  051105
101246  021440  020061  051050
101254  041515  030523  020051
101262     000
101263     105  051122  051117  EMS252: .ASCIZ  @ERROR REGISTER #1 (RMER1) @
101270  051040  043505  051511
101276  042524  020122  030443
101304  024040  046522  051105
101312  024461  000040
101316  051105  047522  020122  EMS253: .ASCIZ  @ERROR REGISTER #2 (RMER2) @
101324  042522  044507  052123
101332  051105  021440  020062
101340  051050  042515  031122
101346  020051     000
101351     115  044501  052116  EMS254: .ASCIZ  @MAINTENANCE REGISTER #1 (RMMR1) @
101356  047105  047101  042503
101364  051040  043505  051511
101372  042524  020122  030443
101400  024040  046522  051115
101406  024461  000040
101412  042504  044523  042522  EMS255: .ASCIZ  @DESIRED CYLINDER REGISTER (RMDC) @
101420  020104  054503  044514
101426  042116  051105  051040
101434  043505  051511  042524
101442  020122  051050  042115
101450  024503  000040
101454  043117  051506  052105  EMS256: .ASCIZ  @OFFSET REGISTER (RMOF) @
101462  051040  043505  051511
101470  042524  020122  051050
101476  047515  024506  000040
101504  051104  053111  020105  EMS257: .ASCIZ  @DRIVE TYPE REGISTER (RMDT) @
101512  054524  042520  051040
101520  043505  051511  042524
101526  020122  051050  042115
101534  024524  000040
101540  047510  042114  047111  EMS260: .ASCIZ  @HOLDING REGISTER (RMHR) @
101546  020107  042522  044507
101554  052123  051105  024040
101562  046522  051110  020051
101570     000
101571     123  051105  040511  EMS261: .ASCIZ  @SERIAL NUMBER REGISTER (RMSN) @
101576  020114  052516  041115
101604  051105  051040  043505
```

```
101612  051511  042524  020122
101620  051050  051515  024516
101626  000040
101630  052101  042524  052116   EMS262: .ASCIZ  @ATTENTION SUMMARY REGISTER (RMAS) @
101636  047511  020116  052523
101644  046515  051101  020131
101652  042522  044507  052123
101660  051105  024040  046522
101666  051501  020051     000

101673     103  047101  047516   EMS300: .ASCIZ  @CANNOT CLEAR @
101700  020124  046103  040505
101706  020122     000
101711     103  047101  047516   EMS301: .ASCIZ  @CANNOT WRITE/READ @
101716  020124  051127  052111
101724  027505  042522  042101
101732  000040
101734  047101  020131  042504   EMS302: .ASCIZ  @ANY DEVICE REGISTER @
101742  044526  042503  051040
101750  043505  051511  042524
101756  020122     000
101761     127  052111  047510   EMS303: .ASCIZ  @WITHOUT @
101766  052125  000040
101772  051105  047522  020122   EMS304: .ASCIZ  @ERROR @
102000     000
102001     101  047440  042516   EMS306: .ASCIZ  @A ONE FROM @
102006  043040  047522  020115
102014     000
102015     125  044523  043516   EMS307: .ASCIZ  @USING MASSBUS INITIALIZE, I.E., @
102022  046440  051501  041123
102030  051525  044440  044516
102036  044524  046101  055111
102044  026105  044440  042456
102052  026056  000040
102056  020101  042532  047522   EMS310: .ASCIZ  @A ZERO FROM @
102064  043040  047522  020115
102072     000
102073     105  042526  054522   EMS311: .ASCIZ  @EVERY DEVICE REGISTER BIT POSITION @
102100  042040  053105  041511
102106  020105  042522  044507
102114  052123  051105  041040
102122  052111  050040  051517
102130  052111  047511  020116
102136     000
102137     124  042510  043040   EMS312: .ASCIZ  @THE FOLLOWING BITS ARE STUCK @
102144  046117  047514  044527
102152  043516  041040  052111
102160  020123  051101  020105
102166  052123  041525  020113
102174     000
102175     101  051440  044510   EMS313: .ASCIZ  @A SHIFTING ONE BIT FROM @
102202  052106  047111  020107
102210  047117  020105  044502
102216  020124  051106  046517
102224  000040
102226  050101  042520  051101   EMS314: .ASCIZ  @APPEARS STUCK AT ZERO @
```

```
102234  020123  052123  041525
102242  020113  052101  055040
102250  051105  020117     000
102255     101  050120  040505   EMS315: .ASCIZ  @APPEARS STUCK AT ONE @
102262  051522  051440  052524
102270  045503  040440  020124
102276  047117  020105     000
102303     122  043505  051511   EMS316: .ASCIZ  @REGISTER SELECT @
102310  042524  020122  042523
102316  042514  052103  000040
102324  020061  030450  031054   EMS317: .ASCIZ  @1 (1,2,4,8,16) @
102332  032054  034054  030454
102340  024466  000040
102344  020062  030450  031054   EMS320: .ASCIZ  @2 (1,2,4,8,16) @
102352  032054  034054  030454
102360  024466  000040
102364  020064  030450  031054   EMS321: .ASCIZ  @4 (1,2,4,8,16) @
102372  032054  034054  030454
102400  024466  000040
102404  020070  030450  031054   EMS322: .ASCIZ  @8 (1,2,4,8,16) @
102412  032054  034054  030454
102420  024466  000040
102424  046101  020114  047117   EMS323: .ASCIZ  @ALL ONES FROM @
102432  051505  043040  047522
102440  020115     000
102443     101  046114  055040   EMS324: .ASCIZ  @ALL ZEROS FROM @
102450  051105  051517  043040
102456  047522  020115     000
102463     101  020124  042532   EMS325: .ASCIZ  @AT ZERO @
102470  047522  000040
102474  052101  047440  042516   EMS326: .ASCIZ  @AT ONE @
102502  000040
102504  020054  051117  000040   EMS327: .ASCIZ  @, OR @
102512  005015  051503  046440   EMS330: .ASCIZ  <CR><LF>@CS MBA CLRL @
102520  040502  041440  051114
102526  020114     000
102531     103  047101  047516   EMS331: .ASCIZ  @CANNOT READ ZEROS FROM @
102536  020124  042522  042101
102544  055040  051105  051517
102552  043040  047522  020115
102560     000
102561     111  020123  047111   EMS332: .ASCIZ  @IS INCORRECT @
102566  047503  051122  041505
102574  020124     000
102577     111  020123  047516   EMS333: .ASCIZ  @IS NOT SET @
102604  020124  042523  020124
102612     000
102613     111  020123  042523   EMS334: .ASCIZ  @IS SET @
102620  020124     000
102623     123  047510  046125   EMS335: .ASCIZ  @SHOULD NOT BE SET @
102630  020104  047516  020124
102636  042502  051440  052105
102644  000040
102646  044123  052517  042114   EMS336: .ASCIZ  @SHOULD BE SET @
102654  041040  020105  042523
102662  020124     000
```

```
102665     103  047101  047516  EMS337: .ASCIZ  @CANNOT SET @
102672  020124  042523  020124
102700     000
102701     102  041505  052501  EMS340: .ASCIZ  @BECAUSE @
102706  042523  000040
102712  051525  047111  020107  EMS341: .ASCIZ  @USING @
102720     000
102721     104  051125  047111  EMS342: .ASCIZ  @DURING REGISTER TRANSFER @
102726  020107  042522  044507
102734  052123  051105  052040
102742  040522  051516  042506
102750  020122     000
102753     125  042516  050130  EMS343: .ASCIZ  @UNEXPECTED @
102760  041505  042524  020104
102766     000
102767     102  051525  052040  EMS344: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
102774  046511  047505  052125
103002  024040  032060  052040
103010  040522  024520  000040
103016  054502  051040  043505  EMS345: .ASCIZ  @BY REGISTER TRANSFER @
103024  051511  042524  020122
103032  051124  047101  043123
103040  051105  000040
103044  040503  047116  052117  EMS346: .ASCIZ  @CANNOT RESET @
103052  051040  051505  052105
103060  000040
103062  044527  044124  052517  EMS347: .ASCIZ  @WITHOUT SETTING @
103070  020124  042523  052124
103076  047111  020107     000
103103     102  052125  000040  EMS350: .ASCIZ  @BUT @
103110  040527  020123  042522  EMS351: .ASCIZ  @WAS RESET BY @
103116  042523  020124  054502
103124  000040
103126  040527  020123  042523  EMS352: .ASCIZ  @WAS SET BY @
103134  020124  054502  000040
103142  047111  042040  040511  EMS353: .ASCIZ  @IN DIAGNOSTIC MODE @
103150  047107  051517  044524
103156  020103  047515  042504
103164  000040
103166  051511  044440  041516  EMS354: .ASCIZ  @IS INCORRECT ACCORDING TO @
103174  051117  042522  052103
103202  040440  041503  051117
103210  044504  043516  052040
103216  020117     000
103221     103  047101  047516  EMS355: .ASCIZ  @CANNOT INCREMENT @
103226  020124  047111  051103
103234  046505  047105  020124
103242     000
103243     127  051501  047040  EMS356: .ASCIZ  @WAS NOT SET BY @
103250  052117  051440  052105
103256  041040  020131     000
103263     127  051501  047040  EMS357: .ASCIZ  @WAS NOT RESET BY @
103270  052117  051040  051505
103276  052105  041040  020131
103304     000
103305     060  052040  020117  EMS360: .ASCIZ  @0 TO 1 TRANSITION OF @
```

```
16685 103312 020061 051124 047101
16686 103320 044523 044524 047117
16687 103326 047440 020106    000
16688 103333    061 052040 020117 EMS361: .ASCIZ  @1 TO 0 TRANSITION OF @
16689 103340 020060 051124 047101
16690 103346 044523 044524 047117
16691 103354 047440 020106    000
16692 103361    111 020123 047111 EMS362: .ASCIZ  @IS INCONSISTENT @
16693 103366 047503 051516 051511
16694 103374 042524 052116 000040
16695 103402 040503 047116 052117 EMS363: .ASCIZ  @CANNOT READ @
16696 103410 051040 040505 020104
16697 103416    000
16698 103417    124 051505 020124 EMS364: .ASCIZ  @TEST PATTERN IN @
16699 103424 040520 052124 051105
16700 103432 020116 047111 000040
16701 103440 047101 020104    000 EMS365: .ASCIZ  @AND @
16702 103445    103 047101 047516 EMS366: .ASCIZ  @CANNOT INITIALIZE @
16703 103452 020124 047111 052111
16704 103460 040511 044514 042532
16705 103466 000040
16706 103470 044124 020105 047503 EMS367: .ASCIZ  @THE COMMAND SEQUENCER HAS BEEN CLOCKED @
16707 103476 046515 047101 020104
16708 103504 042523 052521 047105
16709 103512 042503 020122 040510
16710 103520 020123 042502 047105
16711 103526 041440 047514 045503
16712 103534 042105 000040
```

```
103540  042522  042523  020124  EMS370: .ASCIZ  @RESET EARLY @
103546  040505  046122  020131
103554     000
103555     104  042111  047040  EMS371: .ASCIZ  @DID NOT RESET ON TIME @
103562  052117  051040  051505
103570  052105  047440  020116
103576  044524  042515  000040
103604  052504  044522  043516  EMS372: .ASCIZ  @DURING COMMAND EXECUTION @
103612  041440  046517  040515
103620  042116  042440  042530
103626  052503  044524  047117
103634  000040
103636  047524  000040          EMS373: .ASCIZ  @TO @
103642  044527  044124  040440  EMS374: .ASCIZ  @WITH ANY COMBINATION OF @
103650  054516  041440  046517
103656  044502  040516  044524
103664  047117  047440  020106
103672     000
103673     102  020131  042522  EMS375: .ASCIZ  @BY READING @
103700  042101  047111  020107
103706     000
103707     102  020131  051127  EMS376: .ASCIZ  @BY WRITING @
103714  052111  047111  020107
103722     000
103723     127  051501  051440  EMS377: .ASCIZ  @WAS SET @
103730  052105  000040
103734  040527  020123  047516  EMS400: .ASCIZ  @WAS NOT SET @
103742  020124  042523  020124
103750     000
103751     104  042111  047040  EMS401: .ASCIZ  @DID NOT SET @
103756  052117  051440  052105
103764  000040
103766  044127  046111  020105  EMS402: .ASCIZ  @WHILE @
103774     000
103775     103  046517  040515  EMS403: .ASCIZ  @COMMAND SEQUENCER DID NOT ABORT @
104002  042116  051440  050505
104010  042525  041516  051105
104016  042040  042111  047040
104024  052117  040440  047502
104032  052122  000040
104036  040527  020123  047516  EMS404: .ASCIZ  @WAS NOT RESET @
104044  020124  042522  042523
104052  020124     000
104055     104  051125  047111  EMS405: .ASCIZ  @DURING @
104062  020107     000
104065     127  051501  051040  EMS406: .ASCIZ  @WAS RESET @
104072  051505  052105  000040
104100  042523  051101  044103  EMS407: .ASCIZ  @SEARCH TIMEOUT @
104106  052040  046511  047505
104114  052125  000040

104120  042011  053105  041511  EMS500: .ASCII  @       DEVICE IS NONEXISTENT,@<CR><LF>
104126  020105  051511  047040
104134  047117  054105  051511
104142  042524  052116  006454
104150     012
```

```
104151     011  042504  044526           .ASCII  @      DEVICE IS SWITCHED TO OTHER PORT@<CR><LF>
104156  042503  044440  020123
104164  053523  052111  044103
104172  042105  052040  020117
104200  052117  042510  020122
104206  047520  052122  005015
104214  052011  040522  051516           .ASCIZ  @      TRANSCEIVER ENABLE SWITCH IS OFF@<CR><LF>
104222  042503  053111  051105
104230  042440  040516  046102
104236  020105  053523  052111
104244  044103  044440  020123
104252  043117  006506  000012
104260  044411  020106  047515  EMS501: .ASCIZ  @      IF MODULE, M7686,@<CR><LF>
104266  052504  042514  020054
104274  033515  034066  026066
104302  005015     000
104305     011  040515  051523  EMS502: .ASCIZ  @      MASSBUS TRANSCEIVER,M5922 OR M5923      @<CR><LF>
104312  052502  020123  051124
104320  047101  041523  044505
104326  042526  026122  032515
104334  031071  020062  051117
104342  046440  034465  031462
104350  006411  000012
104354  041411  020123  047515  EMS503: .ASCIZ  @      CS MODULE,M7684,@<CR><LF>
104362  052504  042514  046454
104370  033067  032070  006454
104376  000012
104400  042011  020123  047515  EMS504: .ASCIZ  @      DS MODULE,M7685,@<CR><LF>
104406  052504  042514  046454
104414  033067  032470  006454
104422  000012
104424  042011  053105  041511  EMS505: .ASCIZ  @      DEVICE IS SWITCHED TO A/B PORT POSITION@<CR><LF>
104432  020105  051511  051440
104440  044527  041524  042510
104446  020104  047524  040440
104454  041057  050040  051117
104462  020124  047520  044523
104470  044524  047117  005015
104476     000
104477     011  042504  044526  EMS506: .ASCIZ  @      DEVICE IS NOT AN RM03, OR@<CR><LF>
104504  042503  044440  020123
104512  047516  020124  047101
104520  051040  030115  026063
104526  047440  006522  000012
104534  042011  053105  041511  EMS507: .ASCIZ  @      DEVICE IS SWITCHED TO PROGRAMMABLE PORT POSITION, OR@<CR><LF>
104542  020105  051511  051440
104550  044527  041524  042510
104556  020104  047524  050040
104564  047522  051107  046501
104572  040515  046102  020105
104600  047520  052122  050040
104606  051517  052111  047511
104614  026116  047440  006522
104622  000012
104624  040411  051523  046525  EMS510: .ASCIZ  @      ASSUMING THE RH CONTROLLER HAS NO FAULT@<CR><LF>
104632  047111  020107  044124
```

```
104640  020105  044122  041440
104646  047117  051124  046117
104654  042514  020122  040510
104662  020123  047516  043040
104670  052501  052114  005015
104676     000
104677     015  004412  051120  EMS511: .ASCII  <CR><LF>@       PROBABLE FAULT(S):@<CR><LF>
104704  041117  041101  042514
104712  043040  052501  052114
104720  051450  035051  005015
104726  024011  047516  020124          .ASCIZ  @       (NOT INCLUDING CABLES OR CONNECTORS)@<CR><LF>
104734  047111  046103  042125
104742  047111  020107  040503
104750  046102  051505  047440
104756  020122  047503  047116
104764  041505  047524  051522
104772  006451  000012

104776  042522  042101  044440  EMS600: .ASCIZ  @READ IN PRESET COMMAND @
105004  020116  051120  051505
105012  052105  041440  046517
105020  040515  042116  000040
105026  043117  051506  052105  EMS601: .ASCIZ  @OFFSET COMMAND @
105034  041440  046517  040515
105042  042116  000040
105046  042522  052524  047122  EMS602: .ASCIZ  @RETURN TO CENTER CENTER COMMAND @
105054  052040  020117  042503
105062  052116  051105  041440
105070  047105  042524  020122
105076  047503  046515  047101
105104  020104     000
105107     122  046105  040505  EMS603: .ASCIZ  @RELEASE COMMAND @
105114  042523  041440  046517
105122  040515  042116  000040
105130  042522  040503  044514  EMS604: .ASCIZ  @RECALIBRATE COMMAND @
105136  051102  052101  020105
105144  047503  046515  047101
105152  020104     000
105155     123  042505  020113  EMS605: .ASCIZ  @SEEK COMMAND @
105162  047503  046515  047101
105170  020104     000
105173     123  040505  041522  EMS606: .ASCIZ  @SEARCH COMMAND @
105200  020110  047503  046515
105206  047101  020104     000
105213     104  052101  020101  EMS607: .ASCIZ  @DATA COMMAND @
105220  047503  046515  047101
105226  020104     000

105231     105  050130  052103  EH1:    .ASCII  @EXPCTD  RECEVD  REGSTR@<CR><LF>
105236  020104  051040  041505
105244  053105  020104  051040
105252  043505  052123  006522
105260     012
105261     123  040524  052524          .ASCIZ  @STATUS  STATUS  ADRESS@
105266  020123  051440  040524
105274  052524  020123  040440
```

```
105302  051104  051505  000123
105310  041040  051501  006505  EH2:    .ASCII  @ BASE@<CR><LF>
105316     012
105317     101  051104  051505  EH3:    .ASCIZ  @ADRESS@
105324  000123
105326  054105  041520  042124  EH5:    .ASCII  @EXPCTD  STUCK@<CR><LF>
105334  020040  051440  052524
105342  045503  005015
105346  042522  052523  052114          .ASCIZ  @RESULT  BIT(S)@
105354  020040  044502  024124
105362  024523     000
105365     105  050130  052103  EH7:    .ASCIZ  @EXPCTD  RECEVD@
105372  020104  051040  041505
105400  053105  000104
105404  047123  050107  052122  EH57:   .ASCII  @SNGPRT  DULPRT  RECEVD  DRVTYP@<CR><LF>
105412  020040  052504  050114
105420  052122  020040  042522
105426  042503  042126  020040
105434  051104  052126  050131
105442  005015
105444  051104  052126  050131          .ASCIZ  @DRVTYP  DRVTYP  DRVTYP  REGADR@
105452  020040  051104  052126
105460  050131  020040  051104
105466  052126  050131  020040
105474  042522  040507  051104
105502     000
105503     105  050130  052103  EH65:   .ASCII  @EXPCTD  RECEVD  TEST@<CR><LF>
105510  020104  051040  041505
105516  053105  020104  052040
105524  051505  006524     012
105531     123  040524  052524          .ASCIZ  @STATUS  STATUS  REGSTR@
105536  020123  051440  040524
105544  052524  020123  051040
105552  043505  052123  000122
105560  054105  041520  042124  EH71:   .ASCII  @EXPCTD  RECEVD   TEST@<CR><LF>
105566  020040  042522  042503
105574  042126  020040  052040
105602  051505  006524     012
105607     123  040524  052524          .ASCIZ  @STATUS  STATUS  PATTRN@
105614  020123  051440  040524
105622  052524  020123  050040
105630  052101  051124  000116
105636  054105  041520  042124  EH115:  .ASCII  @EXPCTD  RECEVD  REGSTR   TEST@<CR><LF>
105644  020040  042522  042503
105652  042126  020040  042522
105660  051507  051124  020040
105666  052040  051505  006524
105674     012
105675     123  040524  052524          .ASCIZ  @STATUS  STATUS  ADRESS  PATTRN@
105702  020123  051440  040524
105710  052524  020123  040440
105716  051104  051505  020123
105724  050040  052101  051124
105732  000116
105734  054105  041520  042124  EH130:  .ASCII  @EXPCTD  RECEVD  REGSTR   TEST   OFFSET@<CR><LF>
105742  020040  042522  042503
```

```
16937  105750  042126  020040  042522
16938  105756  051507  051124  020040
16939  105764  052040  051505  020124
16940  105772  020040  043117  051506
16941  106000  052105  005015
16942  106004  052123  052101  051525          .ASCIZ  @STATUS  STATUS  ADRESS  PATTRN  REGSTR@
16943  106012  020040  052123  052101
16944  106020  051525  020040  042101
16945  106026  042522  051523  020040
16946  106034  040520  052124  047122
16947  106042  020040  042522  051507
16948  106050  051124     000
16949  106053     105  050130  052103  EH132:  .ASCII  @EXPCTD  ACTUAL  REGSTR  OFFSET@<CR><LF>
16950  106060  020104  040440  052103
16951  106066  040525  020114  051040
16952  106074  043505  052123  020122
16953  106102  047440  043106  042523
16954  106110  006524     012
16955  106113     103  052517  052116          .ASCIZ  @COUNT   COUNT   ADRESS  REGSTR@
16956  106120  020040  041440  052517
16957  106126  052116  020040  040440
16958  106134  051104  051505  020123
16959  106142  051040  043505  052123
16960  106150  000122
16961  106152  054105  041520  042124  EH145:  .ASCII  @EXPCTD  ACTUAL  REGSTR  RMER1   RMER2@<CR><LF>
16962  106160  020040  041501  052524
16963  106166  046101  020040  042522
16964  106174  051507  051124  020040
16965  106202  046522  051105  020061
16966  106210  020040  046522  051105
16967  106216  006462     012
16968  106221     103  050115  051105          .ASCIZ  @CMPERR  CMPERR  ADRESS  PATTRN  PATTRN@
16969  106226  020122  041440  050115
16970  106234  051105  020122  040440
16971  106242  051104  051505  020123
16972  106250  050040  052101  051124
16973  106256  020116  050040  052101
16974  106264  051124  000116
16975
16976  106270  054105  041520  042124  EH150:  .ASCII  @EXPCTD  ACTUAL  REGSTR FUNCTION@<CR><LF>
16977  106276  020040  041501  052524
16978  106304  046101  020040  042522
16979  106312  051507  051124  043040
16980  106320  047125  052103  047511
16981  106326  006516     012
16982  106331     122  051505  046125          .ASCIZ  @RESULT  RESULT  ADRESS   CODE@
16983  106336  020124  051040  051505
16984  106344  046125  020124  040440
16985  106352  051104  051505  020123
16986  106360  020040  047503  042504
16987  106366     000
16988  106367     105  050130  052103  EH213:  .ASCII  @EXPCTD  ACTUAL  STATUS   TEST@<CR><LF>
16989  106374  020104  040440  052103
16990  106402  040525  020114  051440
16991  106410  040524  052524  020123
16992  106416  020040  042524  052123
```

```
106424  005015
106426  042522  052523  052114              .ASCIZ  @RESULT  RESULT  ADRESS  REGSTR@
106434  020040  042522  052523
106442  052114  020040  042101
106450  042522  051523  020040
106456  042522  051507  051124
106464     000
106465     101  052103  040525  EH220:  .ASCII  @ACTUAL  REGSTR@<CR><LF>
106472  020114  051040  043505
106500  052123  006522     012
106505     122  051505  046125          .ASCIZ  @RESULT  ADRESS@
106512  020124  040440  051104
106520  051505  000123
                                .EVEN
106524  001140  001142  001136  ED1:    .WORD   $GDDAT,$BDDAT,$BDADR,0
106532  000000
106534  001136  000000          ED2:    .WORD   $BDADR,0
106540  001140  001142  000000  ED5:    .WORD   $GDDAT,$BDDAT,0
106546  001174  001176  001142  ED57:   .WORD   $TMP0,$TMP1,$BDDAT,$BDADR,0
106554  001136  000000
106560  001140  001142  001174  ED65:   .WORD   $GDDAT,$BDDAT,$TMP0,0
106566  000000
106570  001140  001142  001440  ED71:   .WORD   $GDDAT,$BDDAT,RMHR0,0
106576  000000
106600  001140  001142  001136  ED115:  .WORD   $GDDAT,$BDDAT,$BDADR,$TMP0,0
106606  001174  000000
106612  001140  001142  001136  ED130:  .WORD   $GDDAT,$BDDAT,$BDADR,$TMP0,$TMP1,0
106620  001174  001176  000000
106626  001142  001136          ED220:  .WORD   $BDDAT,$BDADR

106632     000     000     000  EF1:    .BYTE   0,0,0
106635     000                  EF2:    .BYTE   0
106636     000     000          EF5:    .BYTE   0,0
106640     000     000     000  EF57:   .BYTE   0,0,0,0
106643     000
106644     000     000     000  EF130:  .BYTE   0,0,0,0,0
106647     000     000

        106652                  .EVEN
106652                          BUFFER:
106652  000402                  BUFONE: .BLKW   258.
107656  000402                  BUFTWO: .BLKW   258.
        106652                  .       =       BUFFER
106652                          HELP:
106652  005015                  .ASCII  <CR><LF>
106654  044514  052123  047440  .ASCII  @LIST OF TESTS@<CR><LF>
106662  020106  042524  052123
106670  006523     012
106673     055  026455  026455  .ASCII  @-------------@<CR><LF>
106700  026455  026455  026455
106706  026455  005015
106712  005015                  .ASCII  <CR><LF>
106714  030524  052011  040522  .ASCII  @T1     TRANSFER TEST@<CR><LF>
106722  051516  042506  020122
106730  042524  052123  005015
106736  031124  041411  047524  .ASCII  @T2     CTOD TEST@<CR><LF>
```

```
106744  020104  042524  052123
106752  005015
106754  031524  046411  051501  .ASCII  @T3     MASSBUS INITIALIZE TEST@<CR><LF>
106762  041123  051525  044440
106770  044516  044524  046101
106776  055111  020105  042524
107004  052123  005015
107010  032124  041411  042514  .ASCII  @T4     CLEAR STUCK ACTIVE TEST@<CR><LF>
107016  051101  051440  052524
107024  045503  040440  052103
107032  053111  020105  042524
107040  052123  005015
107044  032524  052011  044522  .ASCII  @T5     TRISTATE TRANSFER TEST@<CR><LF>
107052  052123  052101  020105
107060  051124  047101  043123
107066  051105  052040  051505
107074  006524     012
107077     124  004466  042522  .ASCII  @T6     REGISTER SELECT TEST@<CR><LF>
107104  044507  052123  051105
107112  051440  046105  041505
107120  020124  042524  052123
107126  005015
107130  033524  042011  044522  .ASCII  @T7     DRIVE TYPE TEST@<CR><LF>
107136  042526  052040  050131
107144  020105  042524  052123
107152  005015
107154  030524  004460  042504  .ASCII  @T10    DEVICE AVAILABLE TEST@<CR><LF>
107162  044526  042503  040440
107170  040526  046111  041101
107176  042514  052040  051505
107204  006524     012
107207     124  030461  044011  .ASCII  @T11    HOLDING REGISTER TRANSFER TEST@<CR><LF>
107214  046117  044504  043516
107222  051040  043505  051511
107230  042524  020122  051124
107236  047101  043123  051105
107244  052040  051505  006524
107252     012
107253     124  031061  041411  .ASCII  @T12    CONTROL STATUS #1 TRANSFER TEST@<CR><LF>
107260  047117  051124  046117
107266  051440  040524  052524
107274  020123  030443  052040
107302  040522  051516  042506
107310  020122  042524  052123
107316  005015
107320  030524  004463  051105  .ASCII  @T13    ERROR REGISTER 1 TRANSFER TEST@<CR><LF>
107326  047522  020122  042522
107334  044507  052123  051105
107342  030440  052040  040522
107350  051516  042506  020122
107356  042524  052123  005015
107364  030524  004464  051105  .ASCII  @T14    ERROR REGISTER 2 TRANSFER TEST@<CR><LF>
107372  047522  020122  042522
107400  044507  052123  051105
107406  031040  052040  040522
107414  051516  042506  020122
```

```
107422  042524  052123  005015
107430  030524  004465  046103  .ASCII  @T15    CLEAR OFFSET STUCK ACTIVE TEST@<CR><LF>
107436  040505  020122  043117
107444  051506  052105  051440
107452  052524  045503  040440
107460  052103  053111  020105
107466  042524  052123  005015
107474  030524  004466  043117  .ASCII  @T16    OFFSET REGISTER TRANSFER TEST@<CR><LF>
107502  051506  052105  051040
107510  043505  051511  042524
107516  020122  051124  047101
107524  043123  051105  052040
107532  051505  006524     012
107537     124  033461  051411  .ASCII  @T17    SERIAL NUMBER TEST@<CR><LF>
107544  051105  040511  020114
107552  052516  041115  051105
107560  052040  051505  006524
107566     012
107567     124  030062  041411  .ASCII  @T20    CONTROL BUS PARITY DETECTION TEST@<CR><LF>
107574  047117  051124  046117
107602  041040  051525  050040
107610  051101  052111  020131
107616  042504  042524  052103
107624  047511  020116  042524
107632  052123  005015
107636  031124  004461  047503  .ASCII  @T21    CONTROL BUS PARITY GENERATION TEST@<CR><LF>
107644  052116  047522  020114
107652  052502  020123  040520
107660  044522  054524  043440
107666  047105  051105  052101
107674  047511  020116  042524
107702  052123  005015
107706  031124  004462  046522  .ASCII  @T22    RMDA,RMDC FAULT TEST@<CR><LF>
107714  040504  051054  042115
107722  020103  040506  046125
107730  020124  042524  052123
107736  005015
107740  031124  004463  044504  .ASCII  @T23    DISK ADDRESS TRANSFER TEST@<CR><LF>
107746  045523  040440  042104
107754  042522  051523  052040
107762  040522  051516  042506
107770  020122  042524  052123
107776  005015
110000  031124  004464  042504  .ASCII  @T24    DESIRED CYLINDER TRANSFER TEST@<CR><LF>
110006  044523  042522  020104
110014  054503  044514  042116
110022  051105  052040  040522
110030  051516  042506  020122
110036  042524  052123  005015
110044  031124  004465  046111  .ASCII  @T25    ILLEGAL REGISTER TEST@<CR><LF>
110052  042514  040507  020114
110060  042522  044507  052123
110066  051105  052040  051505
110074  006524     012
110077     124  033062  051011  .ASCII  @T26    RESET GO BY INIT TEST@<CR><LF>
110104  051505  052105  043440
```

```
17161  110112  020117  054502  044440
17162  110120  044516  020124  042524
17163  110126  052123  005015
17164  110132  031124  004467  044504  .ASCII  @T27    DIAGNOSTIC MODE TEST@<CR><LF>
17165  110140  043501  047516  052123
17166  110146  041511  046440  042117
17167  110154  020105  042524  052123
17168  110162  005015
17169  110164  031524  004460  047515  .ASCII  @T30    MOL TEST@<CR><LF>
17170  110172  020114  042524  052123
17171  110200  005015
17172  110202  031524  004461  051127  .ASCII  @T31    WRITE LOCK TEST@<CR><LF>
17173  110210  052111  020105  047514
17174  110216  045503  052040  051505
17175  110224  006524     012
17176  110227     124  031063  042011  .ASCII  @T32    DRIVE FAULT TEST@<CR><LF>
17177  110234  044522  042526  043040
17178  110242  052501  052114  052040
17179  110250  051505  006524     012
17180  110255     124  031463  051411  .ASCII  @T33    SEEK ERROR TEST@<CR><LF>
17181  110262  042505  020113  051105
17182  110270  047522  020122  042524
17183  110276  052123  005015
17184  110302  031524  004464  044520  .ASCII  @T34    PIP TEST@<CR><LF>
17185  110310  020120  042524  052123
17186  110316  005015
17187  110320  031524  004465  041105  .ASCII  @T35    EBL TEST@<CR><LF>
17188  110326  020114  042524  052123
17189  110334  005015
17190  110336  031524  004466  040514  .ASCII  @T36    LAST SECTOR, LAST TRACK TEST@<CR><LF>
17191  110344  052123  051440  041505
17192  110352  047524  026122  046040
17193  110360  051501  020124  051124
17194  110366  041501  020113  042524
17195  110374  052123  005015
17196  110400  031524  004467  046522  .ASCII  @T37    RMDA COUNT TEST@<CR><LF>
17197  110406  040504  041440  052517
17198  110414  052116  052040  051505
17199  110422  006524     012
17200  110425     124  030064  051011  .ASCII  @T40    RMDC COUNT TEST@<CR><LF>
17201  110432  042115  020103  047503
17202  110440  047125  020124  042524
17203  110446  052123  005015
17204  110452  032124  004461  041114  .ASCII  @T41    LBT TEST@<CR><LF>
17205  110460  020124  042524  052123
17206  110466  005015
17207  110470  032124  004462  047503  .ASCII  @T42    COMPOSITE ERROR TEST@<CR><LF>
17208  110476  050115  051517  052111
17209  110504  020105  051105  047522
17210  110512  020122  042524  052123
17211  110520  005015
17212  110522  032124  004463  051127  .ASCII  @T43    WRITE GO TEST@<CR><LF>
17213  110530  052111  020105  047507
17214  110536  052040  051505  006524
17215  110544     012
17216  110545     124  032064  041011  .ASCII  @T44    BRANCH MULTIPLEXOR TEST@<CR><LF>
```

```
110552  040522  041516  020110
110560  052515  052114  050111
110566  042514  047530  020122
110574  042524  052123  005015
110602  032124  004465  042523   .ASCII  @T45    SET/RESET GO TEST@<CR><LF>
110610  027524  042522  042523
110616  020124  047507  052040
110624  051505  006524     012
110631     124  033064  042411   .ASCII  @T46    END 1 RESET GO TEST@<CR><LF>
110636  042116  030440  051040
110644  051505  052105  043440
110652  020117  042524  052123
110660  005015
110662  032124  004467  042523   .ASCII  @T47    SET PULSE TEST@<CR><LF>
110670  020124  052520  051514
110676  020105  042524  052123
110704  005015
110706  032524  004460  042523   .ASCII  @T50    SET/RESET IVC TEST@<CR><LF>
110714  027524  042522  042523
110722  020124  053111  020103
110730  042524  052123  005015
110736  032524  004461  042523   .ASCII  @T51    SET LSC TEST@<CR><LF>
110744  020124  051514  020103
110752  042524  052123  005015
110760  032524  004462  042504   .ASCII  @T52    DECODE TEST@<CR><LF>
110766  047503  042504  052040
110774  051505  006524     012
111001     124  031465  051411   .ASCII  @T53    SET/RESET VOLUME VALID TEST@<CR><LF>
111006  052105  051057  051505
111014  052105  053040  046117
111022  046525  020105  040526
111030  044514  020104  042524
111036  052123  005015
111042  032524  004464  046111   .ASCII  @T54    ILLEGAL FUNCTION TEST@<CR><LF>
111050  042514  040507  020114
111056  052506  041516  044524
111064  047117  052040  051505
111072  006524     012
111075     124  032465  047411   .ASCII  @T55    OCCUPIED TEST@<CR><LF>
111102  041503  050125  042511
111110  020104  042524  052123
111116  005015
111120  032524  004466  042522   .ASCII  @T56    READ IN PRESET TEST@<CR><LF>
111126  042101  044440  020116
111134  051120  051505  052105
111142  052040  051505  006524
111150     012
111151     124  033465  051011   .ASCII  @T57    RIP/RMOF TEST@<CR><LF>
111156  050111  051057  047515
111164  020106  042524  052123
111172  005015
111174  033124  004460  046522   .ASCII  @T60    RMDA/RMDC/RIP TEST@<CR><LF>
111202  040504  051057  042115
111210  027503  044522  020120
111216  042524  052123  005015
111224  033124  004461  043117   .ASCII  @T61    OFFSET COMMAND TEST@<CR><LF>
```

```
111232  051506  052105  041440
111240  046517  040515  042116
111246  052040  051505  006524
111254     012
111255     124  031066  051011   .ASCII  @T62    RETURN TO CENTER TEST@<CR><LF>
111262  052105  051125  020116
111270  047524  041440  047105
111276  042524  020122  042524
111304  052123  005015
111310  033124  004463  046522   .ASCII  @T63    RMDC CLEAR OFFSET TEST@<CR><LF>
111316  041504  041440  042514
111324  051101  047440  043106
111332  042523  020124  042524
111340  052123  005015
111344  033124  004464  041105   .ASCII  @T64    EBL CLEAR OFFSET TEST@<CR><LF>
111352  020114  046103  040505
111360  020122  043117  051506
111366  052105  052040  051505
111374  006524     012
111377     124  032466  051011   .ASCII  @T65    RUN AND GO TEST@<CR><LF>
111404  047125  040440  042116
111412  043440  020117  042524
111420  052123  005015
111424  033124  004466  042523   .ASCII  @T66    SET IAE TEST@<CR><LF>
111432  020124  040511  020105
111440  042524  052123  005015
111446  033124  004467  042523   .ASCII  @T67    SEARCH, SEEK, READ WRITE TEST@<CR><LF>
111454  051101  044103  020054
111462  042523  045505  020054
111470  042522  042101  053440
111476  044522  042524  052040
111504  051505  006524     012
111511     124  030067  044411   .ASCII  @T70    INVALID SECTOR/TRACK TEST@<CR><LF>
111516  053116  046101  042111
111524  051440  041505  047524
111532  027522  051124  041501
111540  020113  042524  052123
111546  005015
111550  033524  004461  047111   .ASCII  @T71    INVALID CYLINDER TEST@<CR><LF>
111556  040526  044514  020104
111564  054503  044514  042116
111572  051105  052040  051505
111600  006524     012
111603     124  031067  051411   .ASCII  @T72    SET AOE TEST@<CR><LF>
111610  052105  040440  042517
111616  052040  051505  006524
111624     012
111625     124  031467  051411   .ASCII  @T73    SET RMR TEST@<CR><LF>
111632  052105  051040  051115
111640  052040  051505  006524
111646     012
111647     124  032067  050011   .ASCII  @T74    PGM STATUS CHECK@<CR><LF>
111654  046507  051440  040524
111662  052524  020123  044103
111670  041505  006513     012
111675     124  032467  042011   .ASCII  @T75    DPR STATUS CHECK@<CR><LF>
```

```
111702  051120  051440  040524
111710  052524  020123  044103
111716  041505  006513     012
111723     124  033067  050011  .ASCII  @T76    PORT REQUEST TEST, PART 1@<CR><LF>
111730  051117  020124  042522
111736  052521  051505  020124
111744  042524  052123  020054
111752  040520  052122  030440
111760  005015
111762  033524  004467  047520  .ASCII  @T77    PORT REQUEST TEST, PART 2@<CR><LF>
111770  052122  051040  050505
111776  042525  052123  052040
112004  051505  026124  050040
112012  051101  020124  006462
112020     012
112021     124  030061  004460  .ASCII  @T100   PORT REQUEST TEST, PART 3@<CR><LF>
112026  047520  052122  051040
112034  050505  042525  052123
112042  052040  051505  026124
112050  050040  051101  020124
112056  006463     012
112061     124  030061  004461  .ASCII  @T101   RELEASE TEST@<CR><LF>
112066  042522  042514  051501
112074  020105  042524  052123
112102  005015
112104  030524  031060  053411  .ASCII  @T102   WRITE ATA TEST@<CR><LF>
112112  044522  042524  040440
112120  040524  052040  051505
112126  006524     012
112131     124  030061  004463  .ASCII  @T103   RESET ATA BY GO TEST@<CR><LF>
112136  042522  042523  020124
112144  052101  020101  054502
112152  043440  020117  042524
112160  052123  005015
112164  030524  032060  052411  .ASCII  @T104   UNIT READY ATA TEST@<CR><LF>
112172  044516  020124  042522
112200  042101  020131  052101
112206  020101  042524  052123
112214  005015
112216  030524  032460  042411  .ASCII  @T105   ERROR ATA TEST@<CR><LF>
112224  051122  051117  040440
112232  040524  052040  051505
112240  006524     012
112243     124  030061  004466  .ASCII  @T106   REGISTER TRANSFER ATA TEST@<CR><LF>
112250  042522  044507  052123
112256  051105  052040  040522
112264  051516  042506  020122
112272  052101  020101  042524
112300  052123  005015
112304  030524  033460  050011  .ASCII  @T107   P SET ATA TEST@<CR><LF>
112312  051440  052105  040440
112320  040524  052040  051505
112326  006524     012
112331     124  030461  004460  .ASCII  @T110   SET WLE TEST@<CR><LF>
112336  042523  020124  046127
112344  020105  042524  052123
```

```
112352  005015
112354  030524  030461  042411   .ASCII  @T111   EXCEPTION TEST@<CR><LF>
112362  041530  050105  044524
112370  047117  052040  051505
112376  006524     012
112401     124  030461  004462   .ASCII  @T112   SET OPI TEST@<CR><LF>
112406  042523  020124  050117
112414  020111  042524  052123
112422  005015
112424  030524  031461  051011   .ASCII  @T113   RECALIBRATE TEST@<CR><LF>
112432  041505  046101  041111
112440  040522  042524  052040
112446  051505  006524     012
112453     124  030461  004464   .ASCII  @T114   SEEK TEST@<CR><LF>
112460  042523  045505  052040
112466  051505  006524     012
112473     124  030461  004465   .ASCII  @T115   SEARCH TEST@<CR><LF>
112500  042523  051101  044103
112506  052040  051505  006524
112514     012
112515     124  030461  004466   .ASCII  @T116   SEARCH TIMEOUT TEST@<CR><LF>
112522  042523  051101  044103
112530  052040  046511  047505
112536  052125  052040  051505
112544  006524     012
112547     124  030461  004467   .ASCII  @T117   DATA COMMAND TESTS (1)@<CR><LF>
112554  040504  040524  041440
112562  046517  040515  042116
112570  052040  051505  051524
112576  024040  024461  005015
112604  030524  030062  042011   .ASCII  @T120   DATA COMMAND TESTS (2)@<CR><LF>
112612  052101  020101  047503
112620  046515  047101  020104
112626  042524  052123  020123
112634  031050  006451     012
112641     124  031061  004461   .ASCII  @T121   DATA COMMAND TESTS(3)@<CR><LF>
112646  040504  040524  041440
112654  046517  040515  042116
112662  052040  051505  051524
112670  031450  006451     012
112675     015     012           .ASCII  <CR><LF>
112677     015     012           .ASCII  <CR><LF>
112701     123  044527  041524   .ASCII  @SWITCH                  USE@<CR><LF>
112706  004510  004411  051525
112714  006505     012
112717     055  026455  026455   .ASCII  @------          --------------------@<CR><LF>
112724  004455  026411  026455
112732  026455  026455  026455
112740  026455  026455  026455
112746  026455  026455  006455
112754     012
112755     040  030440  004465   .ASCII  @  15            HALT ON ERROR@<CR><LF>
112762  044011  046101  020124
112770  047117  042440  051122
112776  051117  005015
113002  020040  032061  004411   .ASCII  @  14            LOOP ON TEST@<CR><LF>
```

```
17441  113010  047514  050117  047440
17442  113016  020116  042524  052123
17443  113024  005015
17444  113026  020040  031461  004411  .ASCII  @  13          INHIBIT ERROR TYPEOUTS@<CR><LF>
17445  113034  047111  044510  044502
17446  113042  020124  051105  047522
17447  113050  020122  054524  042520
17448  113056  052517  051524  005015
17449  113064  020040  030461  004411  .ASCII  @  11          INHIBIT ITERATIONS@<CR><LF>
17450  113072  047111  044510  044502
17451  113100  020124  052111  051105
17452  113106  052101  047511  051516
17453  113114  005015
17454  113116  020040  030061  004411  .ASCII  @  10          BELL ON ERROR@<CR><LF>
17455  113124  042502  046114  047440
17456  113132  020116  051105  047522
17457  113140  006522     012
17458  113143     040  020040  004471  .ASCII  @   9          LOOP ON ERROR@<CR><LF>
17459  113150  046011  047517  020120
17460  113156  047117  042440  051122
17461  113164  051117  005015
17462  113170  020040  034040  004411  .ASCII  @   8          LOOP ON TEST IN SWR<7:0>@<CR><LF>
17463  113176  047514  050117  047440
17464  113204  020116  042524  052123
17465  113212  044440  020116  053523
17466  113220  036122  035067  037060
17467  113226  005015
17468  113230  020040  033440  004411  .ASCII  @   7          TN128@<CR><LF>
17469  113236  047124  031061  006470
17470  113244     012
17471  113245     040  020040  004466  .ASCII  @   6          TN64@<CR><LF>
17472  113252  052011  033116  006464
17473  113260     012
17474  113261     040  020040  004465  .ASCII  @   5          TN32@<CR><LF>
17475  113266  052011  031516  006462
17476  113274     012
17477  113275     040  020040  004464  .ASCII  @   4          TN16@<CR><LF>
17478  113302  052011  030516  006466
17479  113310     012
17480  113311     040  020040  004463  .ASCII  @   3          TN8@<CR><LF>
17481  113316  052011  034116  005015
17482  113324  020040  031040  004411  .ASCII  @   2          TN4@<CR><LF>
17483  113332  047124  006464     012
17484  113337     040  020040  004461  .ASCII  @   1          TN2@<CR><LF>
17485  113344  052011  031116  005015
17486  113352  020040  030040  004411  .ASCII  @   0          TN1@<CR><LF>
17487  113360  047124  006461     012
17488  113365     015  000012          .ASCIZ  <CR><LF>
17489          000001                  .END
```

```
ABASE = 176700     3119#   3227    3268
ACDW1 = 000000     3227    3270
ACDW2 = 000000     3227    3271
ACPUOP= 000000     3227    3242
ADDW0 = 000000     3227    3272
ADDW1 = 000000     3227    3273
ADDW10= 000000     3227
ADDW11= 000000     3227
ADDW12= 000000     3227
ADDW13= 000000     3227
ADDW14= 000000     3227
ADDW15= 000000     3227
ADDW2 = 000000     3227    3274
ADDW3 = 000000     3227    3275
ADDW4 = 000000     3227    3276
ADDW5 = 000000     3227    3277
ADDW6 = 000000     3227    3278
ADDW7 = 000000     3227    3279
ADDW8 = 000000     3227
ADDW9 = 000000     3227
ADEVCT= 000000     3227    3233
ADEVM = 000000     3227    3269
AENV  = 000000     3227    3238
AENVM = 000000     3227    3239
AFATAL= 000000     3227    3230
AMADR1= 000000     3227    3255
AMADR2= 000000     3227    3259
AMADR3= 000000     3227    3262
AMADR4= 000000     3227    3265
AMAMS1= 000000     3227    3249
AMAMS2= 000000     3227    3257
AMAMS3= 000000     3227    3260
AMAMS4= 000000     3227    3263
AMSGAD= 000000     3227    3235
AMSGLG= 000000     3227    3236
AMSGTY= 000000     3227    3229
AMTYP1= 000000     3227    3250
AMTYP2= 000000     3227    3258
AMTYP3= 000000     3227    3261
AMTYP4= 000000     3227    3264
AOE   = 001000     2896#   2907    6204    6235    9241    10118   10122   15080   15081   15084   15085   15088   15089
APASS = 000000     3227    3232
APE   = 100000     3098#
APRIOR= 000000     3227
APTCSU= 000040     14268   14970#
APTENV= 000001     14261   14504   14926   14968#
APTSIZ= 000200      4985   14967#
APTSPO= 000100     14263   14928   14969#
ASWREG= 000000     3227    3240
ATA   = 100000     2876#   10512   10563   10604   10608   10618   10622   10659   10661   10700   10722   10776   10780
                   11262   11264   11632   11634   12045   12047   15061   15062   15063   15066   15067   15070   15071
                   15072   15073   15074   15075   15076   15077   15078   15079   15082   15083   15086   15087   15090
                   15091
ATESTN= 000000     3227    3231
ATNMSK= 000377     2913#   10506
ATNTBL  066064     5155    5173    10505   15093#
```

```
AUNIT  = 000000     3227     3234
AUSWR  = 000000     3227     3241
AVECT1= 120254      3120#    3227     3266
AVECT2= 000000      3227     3267
A16    = 000400     3069#    6981
A17    = 001000     3068#    6981
BAI    = 000010     3088#
BB00   = 000001     3005#
BB01   = 000002     3004#
BB02   = 000004     3003#    13525
BB03   = 000010     3002#    13521    13524
BB04   = 000020     3001#
BB05   = 000040     3000#
BB06   = 000100     2999#    11393    11394    11395    11396    11397    11398    11399    11400    11401    11402
BB07   = 000200     2998#
BB08   = 000400     2997#
BB09   = 001000     2996#
BIT0   = 000001     2792#    5540     6039     6129     6233     6241     6331     6338     6345     6362     6369     6385     6490
                    6497     6513     6778     6846     6858     6984     7077     7145     7231     7316     7443     7529     7603
                    7617     8664
BIT00 = 000001      2782#    2792     2818     2867     2886     2905     2943     2962     3005     3092     3108
BIT01 = 000002      2781#    2791     2817     2866     2904     2942     2961     3004     3091     3107
BIT02 = 000004      2780#    2790     2816     2865     2903     2941     2960     3003     3090     3106
BIT03 = 000010      2779#    2789     2815     2864     2902     2940     2959     3002     3017     3088     3105
BIT04 = 000020      2778#    2788     2814     2863     2901     2958     3001     3087
BIT05 = 000040      2777#    2787     2813     2900     2939     2957     3000     3086
BIT06 = 000100      2776#    2786     2885     2899     2921     2938     2956     2999     3071     3085     3104
BIT07 = 000200      2775#    2785     2884     2898     2920     2937     2955     2979     2998     3016     3070     3084
BIT08 = 000400      2774#    2784     2862     2883     2897     2919     2936     2954     2997     3069     3082     14347
BIT09 = 001000      2773#    2783     2861     2882     2896     2918     2935     2953     2996     3068     3081     14365    14515
BIT1   = 000002     2791#    5597     6054     6145     6265     6394     6522     6793     6872     6984     7090     7096     7158
                    7244     7349     7458     7540     7630
BIT10 = 002000      2772#    2860     2881     2895     2917     2934     2952     2978     2995     3015     3067     3080     3103
                    14492
BIT11 = 004000      2771#    2812     2880     2894     2933     2951     2969     2977     2994     3014     3079     3102     14372
BIT12 = 010000      2770#    2879     2893     2932     2950     2976     2993     3013     3078     3101
BIT13 = 020000      2769#    2878     2892     2931     2949     2968     2992     3012     3065     3077     3100     14499
BIT14 = 040000      2768#    2877     2891     2930     2948     2967     2991     3011     3023     3064     3076     3099     14333
BIT15 = 100000      2767#    2876     2890     2929     2947     2966     2990     3010     3022     3063     3075     3098
BIT2   = 000004     2790#
BIT3   = 000010     2789#
BIT4   = 000020     2788#
BIT5   = 000040     2787#
BIT6   = 000100     2786#
BIT7   = 000200     2785#    5024
BIT8   = 000400     2784#
BIT9   = 001000     2783#
BOTADR   060054     13765*   13783*   13786    13801    13846#
BOTFLG   060056     13751*   13793*   13796    13799*   13847#
BPTVEC= 000014      2799#
BSE    = 100000     3010#
BUFFER   106652     10096    13409    17032#   17035
BUFONE   106652     17033#
BUFTWO   107656     17034#
CC     = 004000     2994#    11393    11394    11395    11396    11397    11398    11399    11400    11401    11402    11782    11783
                    11784    11785    11786    11787    11788    11789    11790    11791    12289    12290    12291    12292    12293
```

```
                   12294   12295   12296   12297   12298   12651   13518   13586   13587   13588   13589   13590   13591
                   13592   13593   13594   13595   13629   13630   13631   13632   13633   13634   13635
CH      = 002000    2995#  11393   11394   11395   11396   11397   11398   11399   11400   11401   11402   12845   12846
                   12847   12848   12849   12850   12851   12852   12853   12854   13516   13597   13598   13599   13600
                   13601   13602   13603   13604   13605   13606   13629   13630   13631   13632   13633   13634   13635
CHRCNT    060057   13752*  13770*  13776*  13777   13780*  13784   13789*  13800*  13848#
CKSWR   = 104410   14332   14488   14514   14865#
CLOCK     001526    3361#   8866    9687   10960   12391   12685   12777   12920   12978   13090   13173   13285   13463
                   13872*  13885*  13896*
CLR     = 000040    3086#   5252    5273    5335    5388    5398    5436    5478    5543    5604    5726    5749    5781
                    5813    5837    5869    5892    5914    5993    6020    6098    6104    6191    6273    6321    6403
                    6442    6564    6610    6683    6720    6763    6832    6919    7020    7026    7057    7131    7221
                    7297    7426    7512    7596    7679    7779    7836    7903    7924    7990    8048    8082    8108
                    8158    8211    8354    8528    8623    8779    8845    9002    9076    9277    9344    9395    9672
                   10147   10220   10254   10492   10548   10589   10644   10686   10922   11016   13968
CLSPRN    065132    5076    5092    5113   14974#
CMNSTA    006052    5030    5162#
CNSL00    065272    5040   14995#
CNSL01    065331    5073   15001#
CNSL02    065356    5082   15005#
CNSL03    065437    5087   15014#
CNSL04    065467    5098   15019#
CNSL05    065543    5108   15028#
CNSL06    065577    5119   15033#
CNSL07    065624    5132   15037#
CONT    = 000100    2956#   8636    8662    8666
CR      = 000015    2707#   5145   13768   14307   14317   14976   14979   14985   14995   15001   15005   15009   15014
                   15019   15023   15028   15033   15038   15048   16606   16764   16769   16775   16781   16785   16792
                   16796   16800   16808   16813   16823   16831   16835   16873   16882   16886   16895   16907   16915
                   16923   16935   16949   16961   16976   16988   17000   17037   17038   17041   17044   17045   17048
                   17051   17056   17061   17066   17071   17075   17080   17087   17094   17100   17106   17112   17118
                   17123   17130   17137   17142   17148   17154   17159   17164   17169   17172   17176   17180   17184
                   17187   17190   17196   17200   17204   17207   17212   17216   17221   17225   17230   17234   17238
                   17241   17244   17250   17255   17259   17264   17268   17272   17277   17282   17287   17292   17296
                   17299   17305   17311   17316   17320   17324   17328   17332   17338   17344   17350   17354   17358
                   17363   17368   17372   17378   17382   17386   17390   17394   17398   17401   17405   17410   17415
                   17420   17425   17426   17427   17430   17436   17440   17444   17449   17454   17458   17462   17468
                   17471   17474   17477   17480   17482   17484   17486   17488
CRLF    = 000200    2708#   5013   14278   14317
CYLMSK  = 001777    2984#   5343
DBCK    = 100000    2929#   8246    8401    8559    8657    8804    8910    8937    8964    9103    9170    9231    9300
                    9362    9415    9476    9535    9758    9854    9921   10023   10105   10449   10768   10836   10839
                   10951   11037   11098   11109   11150   11162   11200   11240   11254   11294   11306   11327   11381
                   11453   11464   11510   11522   11565   11610   11624   11669   11681   11702   11763   11842   11853
                   11899   11911   11957   12003   12020   12037   12082   12094   12115   12158   12175   12188   12206
                   12267   12347   12361   12371   12451   12463   12506   12518   12549   12581   12623   12643   12708
                   12720   12797   12821   12940   12998   13010   13019   13031   13110   13125   13193   13208   13220
                   13232   13305   13318   13328   13351   13492   13539   13566
DBEN    = 040000    2930#   7793    7800    7802    7845    7850    7852    7933    7938    7940    8020    8022    8163
                    8216    8246    8248    8359    8401    8403    8533    8559    8561    8628    8657    8659    8784
                    8804    8806    8850    8910    8912    8937    8939    8964    8966    9007    9081    9103    9105
                    9163    9170    9172    9222    9231    9233    9282    9300    9302    9349    9362    9364    9400
                    9415    9417    9470    9476    9478    9529    9535    9537    9636    9640    9642    9677    9747
                    9758    9760    9845    9854    9856    9915    9921    9923   10017   10023   10025   10088   10105
                   10107   10112   10114   10152   10162   10443   10449   10451   10557   10760   10768   10770   10827
                   10945   10951   10953   11029   11037   11039   11087   11098   11100   11105   11109   11111   11139
```

```
                    11150   11152   11157   11162   11164   11189   11200   11202   11229   11240   11242   11247   11249
                    11254   11256   11283   11294   11296   11301   11306   11308   11322   11327   11329   11354   11381
                    11383   11438   11453   11455   11460   11464   11466   11495   11510   11512   11517   11522   11524
                    11550   11565   11567   11595   11610   11612   11617   11619   11624   11626   11654   11669   11671
                    11676   11681   11683   11697   11702   11704   11731   11763   11765   11827   11842   11844   11849
                    11853   11855   11884   11899   11901   11906   11911   11913   11942   11957   11959   11988   12003
                    12005   12012   12015   12020   12022   12028   12030   12032   12037   12039   12067   12082   12084
                    12089   12094   12096   12110   12115   12117   12143   12158   12160   12167   12170   12175   12177
                    12188   12190   12201   12206   12208   12235   12267   12269   12332   12347   12349   12354   12356
                    12361   12363   12371   12373   12389   12436   12451   12453   12458   12463   12465   12491   12506
                    12508   12513   12518   12520   12529   12531   12549   12551   12560   12562   12581   12583   12608
                    12623   12625   12643   12645   12672   12708   12710   12715   12720   12722   12731   12733   12764
                    12797   12799   12804   12806   12821   12823   12907   12940   12942   12965   12998   13000   13005
                    13010   13012   13019   13021   13026   13031   13033   13042   13044   13081   13110   13112   13118
                    13120   13125   13127   13164   13193   13195   13201   13203   13208   13210   13215   13220   13222
                    13227   13232   13234   13243   13245   13272   13305   13307   13313   13318   13320   13328   13330
                    13346   13351   13353   13362   13364   13454   13492   13494   13506   13508   13539   13541   13554
                    13556   13566   13568
DBL     = 002000     3103#
DCK     = 100000     2890#    2907    6204    6235
DDISP   = 177570     2714#    3203    4973
DEBL    = 020000     2931#    7609    7623    7800    7850    7938    8020    9640    10112
DISPLA    001156     3203#    4973*   4981*  14387*  14491*
DISPRE    000174     3130#    4981
DLT     = 100000     3075#
DMD     = 000001     2943#    2962#   5444    5456    7061    7068    7073    7083    7086    7088    7104    7138    7149
                     7151     7174    7191    7193    7196    7225    7235    7237    7254    7271    7273    7276    7306
                     7339     7341    7380    7400    7402    7405    7434    7448    7452    7468    7486    7488    7491
                     7520     7522    7533    7549    7565    7567    7570    7607    7609    7611    7621    7623    7639
                     7654     7656    7659    7782    7793    7800    7802    7843    7845    7850    7852    7931    7933
                     7938     7940    8012    8020    8022    8055    8085    8111    8161    8163    8214    8216    8246
                     8248     8357    8359    8401    8403    8531    8533    8559    8561    8626    8628    8657    8659
                     8782     8784    8804    8806    8848    8850    8872    8910    8912    8937    8939    8964    8966
                     9005     9007    9079    9081    9103    9105    9163    9170    9172    9222    9231    9233    9280
                     9282     9300    9302    9347    9349    9362    9364    9398    9400    9415    9417    9470    9476
                     9478     9529    9535    9537    9636    9640    9642    9675    9677    9747    9758    9760    9845
                     9854     9856    9915    9921    9923   10017   10023   10025   10088   10105   10107   10112   10114
                    10150    10152   10162   10443   10449   10451   10495   10501   10551   10557   10592   10601   10615
                    10647    10689   10760   10768   10770   10827   10945   10951   10953   11027   11029   11037   11039
                    11087    11098   11100   11105   11109   11111   11139   11150   11152   11157   11162   11164   11189
                    11200    11202   11229   11240   11242   11247   11249   11254   11256   11283   11294   11296   11301
                    11306    11308   11322   11327   11329   11354   11381   11383   11438   11453   11455   11460   11464
                    11466    11495   11510   11512   11517   11522   11524   11550   11565   11567   11595   11610   11612
                    11617    11619   11624   11626   11654   11669   11671   11676   11681   11683   11697   11702   11704
                    11731    11763   11765   11827   11842   11844   11849   11853   11855   11884   11899   11901   11906
                    11911    11913   11942   11957   11959   11988   12003   12005   12012   12015   12020   12022   12028
                    12030    12032   12037   12039   12067   12082   12084   12089   12094   12096   12110   12115   12117
                    12143    12158   12160   12167   12170   12175   12177   12188   12190   12201   12206   12208   12235
                    12267    12269   12332   12347   12349   12354   12356   12361   12363   12371   12373   12389   12436
                    12451    12453   12458   12463   12465   12491   12506   12508   12513   12518   12520   12529   12531
                    12549    12551   12560   12562   12581   12583   12608   12623   12625   12643   12645   12672   12708
                    12710    12715   12720   12722   12731   12733   12764   12797   12799   12804   12806   12821   12823
                    12907    12940   12942   12965   12998   13000   13005   13010   13012   13019   13021   13026   13031
                    13033    13042   13044   13081   13110   13112   13118   13120   13125   13127   13164   13193   13195
                    13201    13203   13208   13210   13215   13220   13222   13227   13232   13234   13243   13245   13272
                    13305    13307   13313   13318   13320   13328   13330   13346   13351   13353   13362   13364   13454
```

N11


```
                        13492   13494   13506   13508   13539   13541   13554   13556   13566   13568   13971   13973   14006
DPE    = 000010          3017#   6342    6366    6620
DPEHI  = 040000          3099#
DPELO  = 020000          3100#
DPR    = 000400          2883#  10267   10269
DRQ    = 004000          2969#  10227   10461
DRVCLR= 000010           2827#   8281    8704
DRY    = 000200          2884#   8385    8389    8424    8431
DSWR   = 177570          2713#   3202    4972
DTE    = 010000          2893#   2907    6212    6244
DTO    = 010000          2932#  12015   12020   12022   12028   12030   12032   12037   12039   12170   12175   12177   12188
                        12190   12201   12206   12208   12361   12363   12371   12373
DULPRT= 024024           2972#   5967    5970
DVA    = 004000          2812#   5997    5999   10258   10260
DVC    = 000200          3016#   5454    6412    6414    7313    7337    7344    7387    7391
EBL    = 020000          2949#   7600    7614    7626    7628    7644    7646   12534   12539   12569   12736   12741   13047
                        13052   13248   13253   13368   13373
ECH    = 000100          2899#   2907    6204    6235   15080   15081   15088   15089
ECI    = 004000          2977#   5345    6518
ECRC   = 001000          2953#
EDT1     074474          3388    3509    3517    3525    3533    3541    3549    3565    3573    3581    3589    3597    3605
                         3613    3621    3629    3637    3645    3661    3669    3677    3685    3693    3701    3709    3717
                         3725    3733    3741    3749    3765    3773    3781    3789    3797    3821    3829    3869    3877
                         3885    3901    3909    3925    3933    3949    3957    3965    3973    3981    3989    4005    4013
                         4029    4037    4053    4061    4069    4117    4125    4133    4141    4157    4165    4173    4181
                         4197    4205    4221    4229    4237    4245    4253    4269    4285    4293    4301    4309    4325
                         4333    4357    4381    4389    4397    4405    4413    4421    4429    4437    4485    4509    4517
                         4525    4565    4573    4581    4589    4597    4605    4638    4654    4670    4678    4687    4695
                         4703    4711    4719    4728    4736    4745    4753    4761    4769    4777    4785    4793    4802
                         4810    4818    4826    4834    4842    4850    4858    4866    4874    4882    4890    4898    4906
                         4915    4923    4931   16024#
EDT115   074512          3997    4021    4045    4077    4213    4261    4277    4341    4349    4365    4445    4461    4469
                         4477    4493    4501    4614    4622    4630    4646    4662   16031#
EDT130   074514          4085    4093    4189   16032#
EDT132   074516          4101    4109    4149   16033#
EDT2     074476          3396   16025#
EDT220   074520          4533    4541    4549    4557   16034#
EDT5     074500          3420    3428    3437   16026#
EDT57    074502          3757   16027#
EDT65    074504          3805    3813    3893    3917    3941   16028#
EDT71    074506          3837    3845    3853   16029#
EDT74    074510          3861   16030#
ED1      106524         16024   17007#
ED115    106600         16031   16033   17017#
ED130    106612         16032   17019#
ED2      106534         16025   16030   17009#
ED220    106626         16034   17021#
ED5      106540         16026   17010#
ED57     106546         16027   17011#
ED65     106560         16028   17013#
ED71     106570         16029   17015#
EECC   = 000020          2958#
EFT1     074522          3389    3510    3518    3526    3534    3542    3550    3566    3574    3582    3590    3598    3606
                         3614    3622    3630    3638    3646    3662    3670    3678    3686    3694    3702    3710    3718
                         3726    3734    3742    3750    3766    3774    3782    3790    3798    3822    3830    3870    3878
                         3886    3902    3910    3926    3934    3950    3958    3966    3974    3982    3990    4006    4014
```

B12


```
                           4030    4038    4054    4062    4070    4118    4126    4134    4142    4158    4166    4174    4182
                           4198    4206    4222    4230    4238    4246    4254    4270    4286    4294    4302    4310    4326
                           4334    4358    4382    4390    4398    4406    4414    4422    4430    4438    4486    4510    4518
                           4526    4566    4574    4582    4590    4598    4606    4639    4655    4671    4679    4688    4696
                           4704    4712    4720    4729    4737    4746    4754    4762    4770    4778    4786    4794    4803
                           4811    4819    4827    4835    4843    4851    4859    4867    4875    4883    4891    4899    4907
                           4916    4924    4932    16036#
EFT115   074540            3998    4022    4046    4078    4214    4262    4278    4342    4350    4366    4446    4462    4470
                           4478    4494    4502    4615    4623    4631    4647    4663    16043#
EFT130   074542            4086    4094    4190    16044#
EFT132   074544            4102    4110    4150    16045#
EFT2     074524            3397    16037#
EFT220   074546            4534    4542    4550    4558    16046#
EFT5     074526            3421    3429    3438    16038#
EFT57    074530            3758    16039#
EFT65    074532            3806    3814    3894    3918    3942    16040#
EFT71    074534            3838    3846    3854    16041#
EFT74    074536            3862    16042#
EF1      106632            16036   16040   16041   17023#
EF130    106644            16044   17028#
EF2      106635            16037   16042   17024#
EF5      106636            16038   16046   17025#
EF57     106640            16039   16043   16045   17026#
EHT1     074400            3387    3508    3516    3524    3532    3540    3548    3564    3572    3580    3588    3596    3604
                           3612    3620    3628    3636    3644    3660    3668    3676    3684    3692    3700    3708    3716
                           3724    3732    3740    3748    3764    3772    3780    3788    3796    3820    3828    3868    3876
                           3884    3900    3908    3924    3932    3948    3956    3964    3972    3980    3988    4004    4012
                           4028    4036    4052    4060    4068    4116    4124    4132    4140    4156    4164    4172    4180
                           4196    4204    4220    4228    4236    4244    4252    4268    4284    4292    4300    4308    4324
                           4332    4356    4380    4388    4396    4404    4412    4420    4428    4436    4484    4508    4516
                           4524    4564    4572    4580    4588    4596    4604    4637    4653    4669    4677    4686    4694
                           4702    4710    4718    4727    4735    4744    4752    4760    4768    4776    4784    4792    4801
                           4809    4817    4825    4833    4841    4849    4857    4865    4873    4881    4889    4897    4905
                           4914    4922    4930    16008#
EHT115   074440            3996    4020    4044    4076    4468    4476    4645    16016#
EHT130   074444            4084    4092    16017#
EHT132   074450            4100    4108    4148    16018#
EHT145   074454            4188    16019#
EHT150   074460            4212    4260    4276    4340    4348    4364    4444    4460    4621    4629    4661    16020#
EHT2     074404            3395    16009#
EHT213   074464            4492    4500    4613    16021#
EHT220   074470            4532    4540    4548    4556    16022#
EHT5     074410            3419    3427    16010#
EHT57    074420            3756    16012#
EHT65    074424            3804    3812    16013#
EHT7     074414            3436    16011#
EHT71    074430            3836    3844    3852    3892    3916    3940    16014#
EHT74    074434            3860    16015#
EH1      105231            16008   16873#
EH115    105636            16016   16923#
EH130    105734            16017   16935#
EH132    106053            16018   16949#
EH145    106152            16019   16961#
EH150    106270            16020   16976#
EH2      105310            16009   16882#
EH213    106367            16021   16988#
```

C12

EH220	106465	16022	17000#											
EH3	105317	16015	16884#											
EH5	105326	16010	16886#											
EH57	105404	16012	16895#											
EH65	105503	16013	16907#											
EH7	105365	16011	16892#											
EH71	105560	16014	16915#											
EMS1	074550	15245	15246	16048#										
EMS10	075153	15422	15427	16094#										
EMS11	075216	15432	16100#											
EMS12	075307	15404	15407	15410	16110#									
EMS13	075365	15440	15443	15446	15449	16118#								
EMS14	075434	15453	15458	15466	16125#									
EMS15	075501	15453	15458	15463	15581	15813	15817	15855	15860	15892	15924	15963	16132#	
EMS16	075557	15468	15473	15478	15833	16140#								
EMS17	075640	15468	15473	15481	16149#									
EMS2	074621	15253	15545	15588	15639	15647	15656	15678	15839	15850	16055#			
EMS20	075701	15483	15488	15492	15496	15501	15506	15513	16155#					
EMS21	075745	15483	15488	15496	15501	15506	15510	15865	15897	15929	15968	16162#		
EMS22	076024	15488	15492	15501	15506	16170#								
EMS23	076071	15515	15520	15528	16177#									
EMS24	076147	15515	15520	15525	15883	15915	15947	15999	16185#					
EMS25	076226	15530	15535	15543	16193#									
EMS250	101161	15291	15294	15297	15339	15562	15566	15570	15574	15702	15711	15745	15754	15792
		16465#												
EMS251	101217	15299	15302	15305	15765	15784	16471#							
EMS252	101263	15307	15312	15314	15317	15320	15322	15325	15328	15330	15606	15610	15844	16478#
EMS253	101316	15307	15345	15348	15352	15356	15360	15364	15368	15606	15610	16483#		
EMS254	101351	15307	15449	15463	15478	15510	15525	15540	15557	16488#				
EMS255	101412	15332	15334	15337	15339	15578	15592	15702	15715	15731	15745	15758	16494#	
EMS256	101454	15371	15374	15378	15382	15702	15707	16500#						
EMS257	101504	15385	16504#											
EMS26	076304	15530	15535	15540	15870	15902	15934	15994	16201#					
EMS260	101540	15393	15397	15401	15432	16509#								
EMS261	101571	15585	16514#											
EMS262	101630	15765	15788	15800	15805	16520#								
EMS27	076363	15545	15549	15553	15560	15736	15971	15973	15977	16002	16209#			
EMS3	074670	15343	15621	15643	15651	15691	15698	15741	15747	15751	15833	16062#		
EMS30	076427	15551	15553	15557	15570	15574	15592	15599	16216#					
EMS300	101673	15245	15253	15312	15314	15317	15343	15348	15352	15404	15413	15417	15440	15545
		15588	15639	15647	15656	15678	15702	15707	15711	15715	15839	15850	16527#	
EMS301	101711	15246	15250	15257	15261	15264	15291	15294	15297	15299	15302	15305	15307	15320
		15322	15325	15328	15330	15332	15334	15337	15339	15356	15360	15364	15368	15374
		15378	15382	15393	15397	15401	15432	15449	15463	15478	15510	15525	15540	15557
		15614	15694	16530#										
EMS302	101734	15246	15250	15253	15264	16534#								
EMS303	101761	15246	15432	16538#										
EMS304	101772	15246	16540#											
EMS306	102001	15250	15261	15307	15614	15694	15778	15781	16542#					
EMS307	102015	15253	15545	15588	15639	15647	15656	15678	15839	15850	16545#			
EMS31	076507	15562	16225#											
EMS310	102056	15257	15603	15805	16551#									
EMS311	102073	15257	15261	16554#										
EMS312	102137	15260	15293	15296	15301	15304	15333	15336	15358	15362	15366	15376	15380	15395
		15399	16561#											
EMS313	102175	15264	15297	15305	15330	15337	15368	15382	15401	15449	15463	15478	15510	15525

D12


```
                            15540   15557   16567#
EMS314   102226             15267   15273   15279   15285   15311   16572#
EMS315   102255             15270   15276   15282   15288   15488   15551   16576#
EMS316   102303             15267   15270   15273   15276   15279   15282   15285   15288   16580#
EMS317   102324             15267   15270   16583#
EMS32    076560             15566   16232#
EMS320   102344             15273   15276   16586#
EMS321   102364             15279   15282   16589#
EMS322   102404             15285   15288   16592#
EMS323   102424             15294   15302   15328   15334   15339   15364   15378   15397   16595#
EMS324   102443             15291   15299   15320   15322   15325   15332   15356   15360   15374   15393   16598#
EMS325   102463             15296   15304   15336   15366   15380   15399   16601#
EMS326   102474             15293   15301   15333   15358   15362   15376   15395   16603#
EMS327   102504             15307   15339   15417   15606   15672   15702   15745   15747   15765   16605#
EMS33    076640             15570   16240#
EMS330   102512             15311   16606#
EMS331   102531             15345   15371   15549   15578   16609#
EMS332   102561             15385   15466   15481   15513   15528   15543   15560   15824   15888   15920   15956   15990   16614#
EMS333   102577             15389   15415   15422   15492   15506   15654   15675   15683   15774   15784   15788   15792   15805
                            16617#
EMS334   102613             15427   15492   15501   15664   15765   15774   16620#
EMS335   102623             15407   15422   15664   15668   15683   15769   15868   15886   15900   15918   15932   15950   16622#
EMS336   102646             15410   15427   15860   15870   15892   15902   15924   15934   15963   15971   15994   16002   16626#
EMS337   102665             15446   15458   15473   15496   15501   15520   15535   15553   15599   15606   15625   15660   15664
                            15668   15672   15688   15719   15745   15762   15771   16629#
EMS34    076677             15574   15691   16246#
EMS340   102701             15422   15427   15683   15860   15870   15892   15902   15924   15934   15963   15994   16632#
EMS341   102712             15453   15458   15468   15473   15483   15496   15501   15515   15520   15530   15535   15553   15570
                            15574   15592   15599   15606   15704   15707   15711   15715   15719   15723   15727   15736   15796
                            16634#
EMS342   102721             15407   15410   15422   15427   15437   15824   16636#
EMS343   102753             15437   16641#
EMS344   102767             15437   16644#
EMS345   103016             15443   15446   15596   15731   15765   16648#
EMS346   103044             15443   15453   15468   15483   15515   15530   15596   15723   15727   15731   15736   15796   15800
                            16652#
EMS347   103062             15449   15463   15478   15510   15525   15540   15557   16655#
EMS35    076735             15581   15654   15664   15672   15688   16252#
EMS350   103103             15488   15492   15501   15506   15774   15805   15844   15855   16658#
EMS351   103110             15488   16659#
EMS352   103126             15506   16662#
EMS353   103142             15549   15621   15631   15635   15643   15651   15660   15698   15828   16664#
EMS354   103166             15562   15566   15610   15621   15628   15643   15651   15698   15741   15751   15754   15758   15833
                            16668#
EMS355   103221             15570   15574   15592   16673#
EMS356   103243             15813   15817   15821   15960   16677#
EMS357   103263             15581   15809   16680#
EMS36    076777             15588   15614   15625   15628   15631   15635   15683   15809   15868   15879   15886   15900   15911
                            15918   15932   15943   15950   15981   16258#
EMS360   103305             15581   15813   16684#
EMS361   103333             15817   16688#
EMS362   103361             15585   16692#
EMS363   103402             15603   15778   15781   15805   16695#
EMS364   103417             15610   15754   15758   16698#
EMS365   103440             15610   15621   15631   15635   15643   15651   15698   15824   15828   15833   16701#
EMS366   103445             15618   16702#
```

```
EMS367  103470      15621   15631   15635   15643   15651   15698   15828   16706#
EMS37   077027      15592   16263#
EMS370  103540      15631   15879   15911   15943   15973   16713#
EMS371  103555      15635   15977   16716#
EMS372  103604      15664   15668   15675   15809   15833   15844   15855   16720#
EMS373  103636      15731   15765   16725#
EMS374  103642      15745   16726#
EMS375  103673      15784   16731#
EMS376  103707      15788   15792   15800   15844   16734#
EMS377  103723      15824   15833   15844   15865   15883   15897   15915   15929   15947   15968   15999   16737#
EMS4    074735      15345   15371   15578   16069#
EMS40   077067      15596   15599   16269#
EMS400  103734      15855   16739#
EMS401  103751      15828   15846   15855   15875   15907   15939   15952   15985   16004   16742#
EMS402  103766      15865   15883   15897   15915   15929   15947   15968   15999   16745#
EMS403  103775      15865   15883   15897   15915   15929   15947   15968   15999   16747#
EMS404  104036      15870   15902   15934   15981   15994   16753#
EMS405  104055      15860   15870   15875   15879   15888   15892   15902   15907   15911   15920   15924   15934   15939
                    15943   15952   15956   15963   15973   15977   15981   15985   15990   15994   16004   16756#
EMS406  104065      15860   15892   15924   15963   16758#
EMS407  104100      15960   16760#
EMS41   077145      15603   15606   15610   15664   15668   15675   15821   15824   16277#
EMS42   077213      15618   15621   16284#
EMS43   077275      15628   16293#
EMS44   077346      15639   15643   16300#
EMS45   077407      15647   15651   16306#
EMS46   077464      15656   15660   16314#
EMS47   077546      15668   15672   15678   15698   16323#
EMS5    074766      15389   15778   16074#
EMS50   077606      15683   15691   16329#
EMS500  104120      15248   16764#
EMS501  104260      15248   15251   15255   15258   15262   15265   15268   15271   15274   15277   15280   15283   15286
                    15289   15300   15303   15306   15309   15313   15315   15318   15321   15323   15326   15329   15331
                    15341   15344   15346   15349   15353   15357   15361   15365   15369   15372   15375   15379   15383
                    15386   15390   15394   15398   15402   15405   15408   15411   15414   15419   15424   15429   15434
                    15455   15460   15470   15475   15485   15490   15494   15498   15504   15508   15517   15522   15532
                    15537   15583   15590   15604   15608   15612   15615   15626   15629   15653   15661   15666   15670
                    15674   15680   15685   15689   15692   15695   15700   15705   15709   15721   15725   15729   15733
                    15738   15748   15752   15768   15772   15776   15779   15782   15786   15790   15794   15798   15802
                    15807   15811   15815   15819   15822   15826   15830   15836   15847   15852   15857   16781#
EMS502  104305      15248   15251   15255   15258   15262   15265   15268   15271   15274   15277   15280   15283   15286
                    15289   15309   15424   15429   15434   15742   15841   15847   15987   16785#
EMS503  104354      15248   15255   15258   15262   15265   15309   15318   15326   15341   15349   15357   15441   15444
                    15447   15451   15455   15460   15465   15470   15475   15480   15485   15498   15512   15517   15522
                    15527   15532   15537   15542   15547   15550   15555   15559   15563   15567   15586   15615   15619
                    15623   15633   15637   15641   15645   15649   15653   15658   15661   15680   15685   15742   15802
                    15830   15836   15841   15847   15852   15857   15863   15867   15873   15877   15881   15885   15890
                    15895   15899   15905   15909   15913   15917   15922   15927   15931   15937   15941   15945   15949
                    15954   15958   15961   15966   15970   15975   15979   15983   15987   15992   15997   16001   16006
                    16792#
EMS504  104400      15258   15262   15265   15292   15295   15298   15315   15323   15335   15338   15341   15353   15361
                    15419   15563   15567   15572   15576   15579   15594   15597   15601   15695   15713   15717   15733
                    15738   15748   15756   15760   15763   16796#
EMS505  104424      15251   16800#
EMS506  104477      15386   16808#
EMS507  104534      15390   16813#
```

```
EMS51    077655      15694  15727  16336#
EMS510   104624      15742  16823#
EMS511   104677      15248  15251  15255  15258  15262  15265  15268  15271  15274  15277  15280  15283  15286
                     15289  15292  15295  15298  15300  15303  15306  15309  15313  15315  15318  15321  15323
                     15326  15329  15331  15335  15338  15341  15344  15346  15349  15353  15357  15361  15365
                     15369  15372  15375  15379  15383  15386  15390  15394  15398  15402  15405  15408  15411
                     15414  15419  15424  15429  15434  15441  15444  15447  15451  15455  15460  15465  15470
                     15475  15480  15485  15490  15494  15498  15504  15508  15512  15517  15522  15527  15532
                     15537  15542  15547  15550  15555  15559  15563  15567  15572  15576  15579  15583  15586
                     15590  15594  15597  15601  15604  15608  15612  15615  15619  15623  15626  15629  15633
                     15637  15641  15645  15649  15653  15658  15661  15666  15670  15674  15680  15685  15689
                     15692  15695  15700  15705  15709  15713  15717  15721  15725  15729  15733  15738  15742
                     15748  15752  15756  15760  15763  15768  15772  15776  15779  15782  15786  15790  15794
                     15798  15802  15807  15811  15815  15819  15822  15826  15830  15836  15841  15847  15852
                     15857  15863  15867  15873  15877  15881  15885  15890  15895  15899  15905  15909  15913
                     15917  15922  15927  15931  15937  15941  15945  15949  15954  15958  15961  15966  15970
                     15975  15979  15983  15987  15992  15997  16001  16006  16831#
EMS52    077724      15719  15723  15731  15736  16343#
EMS53    077765      15741  15985  16349#
EMS54    100026      15745  15751  15754  15758  16355#
EMS55    100103      15762  16363#
EMS56    100161      15765  15769  15771  15844  16371#
EMS57    100254      15774  16382#
EMS6     075036      15413  15417  15422  15427  16081#
EMS60    100340      15774  16391#
EMS600   104776      15704  15707  15711  15715  16843#
EMS601   105026      15719  16847#
EMS602   105046      15723  15727  16850#
EMS603   105107      15796  16856#
EMS604   105130      15860  15865  15870  15875  15879  15883  15888  16859#
EMS605   105155      15892  15897  15902  15907  15911  15915  15920  16863#
EMS606   105173      15924  15929  15934  15939  15943  15947  15952  15956  16866#
EMS607   105213      15963  15968  15973  15977  15981  15985  15990  15994  15999  16004  16869#
EMS61    100412      15781  16398#
EMS62    100465      15784  15788  15792  15796  16406#
EMS63    100546      15800  15805  15809  15813  15817  15821  15824  15828  15875  15907  15939  16415#
EMS64    100606      15833  16421#
EMS65    100656      15839  15846  16428#
EMS66    100717      15850  15855  15860  15870  15892  15902  15924  15934  15960  15963  15994  16434#
EMS67    101001      15888  15920  15956  15990  16443#
EMS7     075102      15415  15417  16087#
EMS70    101113      15952  16004  16457#
EMTVEC=  000030       2802#  4956*  4957*  5200*
EMT1     066522       3386  15245#
EMT10    066710       3443  15267#
EMT100   070324       3891  15449#
EMT101   070344       3899  15453#
EMT102   070364       3907  15458#
EMT103   070404       3915  15463#
EMT104   070430       3923  15468#
EMT105   070450       3931  15473#
EMT106   070470       3939  15478#
EMT107   070514       3947  15483#
EMT11    066726       3451  15270#
EMT110   070534       3955  15488#
EMT111   070556       3963  15492#
```

EMT112	070576	3971	15496#
EMT113	070616	3979	15501#
EMT114	070642	3987	15506#
EMT115	070664	3995	15510#
EMT116	070710	4003	15515#
EMT117	070730	4011	15520#
EMT12	066744	3459	15273#
EMT120	070750	4019	15525#
EMT121	070774	4027	15530#
EMT122	071014	4035	15535#
EMT123	071034	4043	15540#
EMT124	071060	4051	15545#
EMT125	071076	4059	15549#
EMT126	071116	4067	15553#
EMT127	071134	4075	15557#
EMT13	066762	3467	15276#
EMT130	071160	4083	15562#
EMT131	071176	4091	15566#
EMT132	071214	4099	15570#
EMT133	071234	4107	15574#
EMT134	071254	4115	15578#
EMT135	071270	4123	15581#
EMT136	071306	4131	15585#
EMT137	071320	4139	15588#
EMT14	067000	3475	15279#
EMT140	071336	4147	15592#
EMT141	071356	4155	15596#
EMT142	071372	4163	15599#
EMT143	071410	4171	15603#
EMT144	071424	4179	15606#
EMT145	071446	4187	15610#
EMT146	071470	4195	15614#
EMT147	071506	4203	15618#
EMT15	067016	3483	15282#
EMT150	071520	4211	15621#
EMT151	071542	4219	15625#
EMT152	071554	4227	15628#
EMT153	071570	4235	15631#
EMT154	071610	4243	15635#
EMT155	071630	4251	15639#
EMT156	071646	4259	15643#
EMT157	071670	4267	15647#
EMT16	067034	3491	15285#
EMT160	071706	4275	15651#
EMT161	071736	4283	15656#
EMT162	071754	4291	15660#
EMT163	071772	4299	15664#
EMT164	072014	4307	15668#
EMT165	072036	4315	15672#
EMT166	072062	4323	15678#
EMT167	072102	4331	15683#
EMT17	067052	3499	15288#
EMT170	072124	4339	15688#
EMT171	072136	4347	15691#
EMT172	072152	4355	15694#
EMT173	072170	4363	15698#

EMT174	072212	4371	15702#	
EMT175	072240	4379	15707#	
EMT176	072256	4387	15711#	
EMT177	072274	4395	15715#	
EMT2	066530	3394	15246#	
EMT20	067070	3507	15291#	
EMT200	072312	4403	15719#	
EMT201	072330	4411	15723#	
EMT202	072346	4419	15727#	
EMT203	072364	4427	15731#	
EMT204	072406	4435	15736#	
EMT205	072426	4443	15741#	
EMT206	072446	4451	15745#	
EMT207	072476	4459	15751#	
EMT21	067110	3515	15294#	
EMT210	072512	4467	15754#	
EMT211	072530	4475	15758#	
EMT212	072546	4483	15762#	
EMT213	072560	4491	15765#	
EMT214	072610	4499	15771#	
EMT215	072622	4507	15774#	
EMT216	072642	4515	15778#	
EMT217	072656	4523	15781#	
EMT22	067130	3523	15297#	
EMT220	072672	4531	15784#	
EMT221	072710	4539	15788#	
EMT222	072726	4547	15792#	
EMT223	072744	4555	15796#	
EMT224	072762	4563	15800#	
EMT225	073002	4571	15805#	
EMT226	073024	4579	15809#	
EMT227	073042	4587	15813#	
EMT23	067144	3531	15299#	
EMT230	073060	4595	15817#	
EMT231	073076	4603	15821#	
EMT232	073112	4612	15824#	
EMT233	073134	4620	15828#	
EMT234	073156	4628	15833#	
EMT235	073204	4636	15839#	
EMT236	073224	4644	15844#	
EMT237	073256	4652	4660	15850#
EMT24	067164	3539	15302#	
EMT240	073276	15855#		
EMT241	073322	4668	15860#	
EMT242	073346	4676	15865#	
EMT243	073372	4685	15870#	
EMT244	073416	4693	15875#	
EMT245	073434	4701	15879#	
EMT246	073452	4709	15883#	
EMT247	073476	4717	15888#	
EMT25	067204	3547	15305#	
EMT250	073514	4726	15892#	
EMT251	073540	4734	15897#	
EMT252	073564	4743	15902#	
EMT253	073610	4751	15907#	
EMT254	073626	4759	15911#	

EMT255	073644	4767	15915#
EMT256	073670	4775	15920#
EMT257	073706	4783	15924#
EMT26	067220	3555	15307#
EMT260	073732	4791	15929#
EMT261	073756	4800	15934#
EMT262	074002	4808	15939#
EMT263	074020	4816	15943#
EMT264	074036	4824	15947#
EMT265	074062	4832	15952#
EMT266	074100	4840	15956#
EMT267	074116	4848	15960#
EMT27	067252	3563	15312#
EMT270	074132	4856	15963#
EMT271	074156	4864	15968#
EMT272	074202	4872	15973#
EMT273	074220	4880	15977#
EMT274	074236	4888	15981#
EMT275	074254	4896	15985#
EMT276	074274	4904	15990#
EMT277	074312	4913	15994#
EMT3	066556	3402	15250#
EMT30	067264	3571	15314#
EMT300	074336	4921	15999#
EMT301	074362	4929	16004#
EMT31	067300	3579	15317#
EMT32	067314	3587	15320#
EMT33	067330	3595	15322#
EMT34	067346	3603	15325#
EMT35	067364	3611	15328#
EMT36	067400	3619	15330#
EMT37	067414	3627	15332#
EMT4	066576	3410	15253#
EMT40	067430	3635	15334#
EMT41	067450	3643	15337#
EMT42	067464	3651	15339#
EMT43	067510	3659	15343#
EMT44	067522	3667	15345#
EMT45	067536	3675	15348#
EMT46	067552	3683	15352#
EMT47	067566	3691	15356#
EMT5	066620	3418	15257#
EMT50	067610	3699	15360#
EMT51	067632	3707	15364#
EMT52	067652	3715	15368#
EMT53	067666	3723	15371#
EMT54	067702	3731	15374#
EMT55	067722	3739	15378#
EMT56	067742	3747	15382#
EMT57	067756	3755	15385#
EMT6	066644	3426	15261#
EMT60	067772	3763	15389#
EMT61	070006	3771	15393#
EMT62	070026	3779	15397#
EMT63	070046	3787	15401#
EMT64	070062	3795	15404#

```
EMT65    070074     3803    15407#
EMT66    070110     3811    15410#
EMT67    070124     3819    15413#
EMT7     066666     3435    15264#
EMT70    070142     3827    15417#
EMT71    070162     3835    15422#
EMT72    070206     3843    15427#
EMT73    070232     3851    15432#
EMT74    070252     3859    15437#
EMT75    070262     3867    15440#
EMT76    070274     3875    15443#
EMT77    070310     3883    15446#
ENRGDT   066522    15242#
EQUALS   065133    14975#
ERR    = 040000     2877#    8062     8066     8074     8094     8120     8121     9017
ERRNMB   060052    13732*   13733*   13736    13748    13753    13845#
ERRTYP   057314    13711#   14501
ERRVEC=  000004     2795#    4970     4971*    4982*    6935     6936     6937*    6938*    6944*    6945*    6950*    6951*    6976
                    6977     6978*    6979*    6988*    6989*    7000*    7001*   13865    13866    13867*   13868*   13883*   13904*
                   13905*   14338    14339*   14341*   14344*
ERTY00   060060    13718    13850#
ERTY01   060066    13726    13851#
ERTY02   060076    13735    13853#
ERTY03   060105    13741    13855#
ERTY04   060113    13837    13857#
ESRC   = 004000     2951#   12185    12194    12376    12381    13333    13338
FER    = 000020     2901#    2907     6204     6235
FMT16  = 010000     2976#    5345     6518     7704     7722     7742     7744     7873     7875     7958     7960     9631
FNCDTB   065764     8810     9178     9241     9863    15058#
FNCMSK=  000077     2819#    8178     8237     8374     8406     8549     8564     8645
F0     = 000002     2817#
F1     = 000004     2816#
F2     = 000010     2815#
F3     = 000020     2814#
F4     = 000040     2813#
GNS    = ****** U   3129     5012    13670    13677    14856    14857    14858    14859    14860    14861    14863    14865    14866
                   14867    14868    14869    14870
GO     = 000001     2818#    7025     7030     7795     7847     7935     8018     8170     8175     8236     8367     8372     8383
                    8412     8418     8420     8425     8541     8547     8567     8573     8575     8644     8800     8864     8906
                    8933     8960     9096     9165     9224     9296     9358     9411     9472     9531     9638     9685     9750
                    9847     9917    10019    10100    10159    10300    10347    10394    10445    10560    10762    10832    10947
                   11031    11093    11145    11169    11195    11235    11289    11314    11316    11334    11360    11448    11505
                   11529    11560    11605    11664    11689    11691    11709    11741    11837    11894    11918    11952    11998
                   12077    12102    12104    12122    12153    12213    12245    12342    12446    12501    12588    12618    12682
                   12774    12917    12975    13087    13170    13282    13407    13979    14012
GTSWR  = 104407     5007     5196    14863#
HCE    = 000200     2898#    2907     6204     6235    15080    15081    15084    15088    15089
HCI    = 002000     2978#    5345     6518
HCRC   = 000400     2897#    2907     6204     6235
HELP     106652     5060    17036#
HELPQS   065143     5054    14978#
HT     = 000011     2705#   13774    14276    14317
IAE    = 002000     2895#    6204     6235     9766     9862     9864     9929     9931    10031    10033    15062    15072    15080
                   15081    15084    15085    15088    15089
IDXMSK=  000077     3057#
IE     = 000100     3071#    3104#
```

K12


```
ILF    = 000001     2905#   6196    6227    9044    9177    9179   15061   15070   15071   15073   15074   15075   15076
                   15077   15078   15079   15082   15083   15086   15087   15090   15091
ILF02 = 000002      2824#   8448    8692
ILF24 = 000024      2834#   8463    8728    9041
ILF26 = 000026      2835#   8466    8732
ILF30 = 000030      2837#
ILF32 = 000032      2838#   8321    8595    8740
ILF34 = 000034      2839#   8469    8744
ILF36 = 000036      2840#   8472    8748
ILF40 = 000040      2841#
ILF42 = 000042      2842#   8475
ILF44 = 000044      2843#   8478
ILF46 = 000046      2844#   8481
ILF54 = 000054      2847#   8484
ILF56 = 000056      2848#   8487
ILF64 = 000064      2851#   8490
ILF66 = 000066      2852#   8493
ILF74 = 000074      2855#   8496
ILF76 = 000076      2856#   5339    5360    5392    5408    5482    5519    5555    5580    5820    5827    6103    6108
                    6118    6123    6135    6138    6139    6153    6160    8185    8366    8499    8540    8819    9095
                    9189    9254    9706    9749    9875   11053
ILR    = 000002     2904#   6196    6227    6923    6961    6965    6986    6992
ILRG50= 000050      3043#
ILRG52= 000052      3044#
ILRG54= 000054      3045#
ILRG56= 000056      3046#
ILRG60= 000060      3047#
ILRG62= 000062      3048#
ILRG64= 000064      3049#
ILRG66= 000066      3050#
ILRG70= 000070      3051#
ILRG72= 000072      3052#
ILRG74= 000074      3053#
ILRG76= 000076      3054#
IOTVEC= 000020      2800#   4954*   4955*
IPCK0 = 000001      3108#
IPCK1 = 000002      3107#
IPCK2 = 000004      3106#
IPCK3 = 000010      3105#
IR     = 000100     3085#   5260    5281
IVC    = 010000     3013#   6335    6359    8791    8809    8811   15061   15062   15063   15065   15066   15067   15070
                   15071   15072   15073   15074   15075   15076   15077   15078   15079   15080   15081   15082   15083
                   15084   15085   15086   15087   15088   15089   15090   15091
LBC    = 002000     3015#   6342    6366
LBT    = 002000     2881#   8000    8025    8027
LCLOCK  060374     13885   13914#
LCOUNT  060444     13887   13932#
LF     = 000012     2706#  13772   14311   14317   14976   14979   14985   14995   15001   15005   15009   15014   15019
                   15023   15028   15033   15038   15048   16606   16764   16769   16775   16781   16785   16792   16796
                   16800   16808   16813   16823   16831   16835   16873   16882   16886   16895   16907   16915   16923
                   16935   16949   16961   16976   16988   17000   17037   17038   17041   17044   17045   17048   17051
                   17056   17061   17066   17071   17075   17080   17087   17094   17100   17106   17112   17118   17123
                   17130   17137   17142   17148   17154   17159   17164   17169   17172   17176   17180   17184   17187
                   17190   17196   17200   17204   17207   17212   17216   17221   17225   17230   17234   17238   17241
                   17244   17250   17255   17259   17264   17268   17272   17277   17282   17287   17292   17296   17299
                   17305   17311   17316   17320   17324   17328   17332   17338   17344   17350   17354   17358   17363
```

```
                         17368   17372   17378   17382   17386   17390   17394   17398   17401   17405   17410   17415   17420
                         17425   17426   17427   17430   17436   17440   17444   17449   17454   17458   17462   17468   17471
                         17474   17477   17480   17482   17484   17486   17488
LS       = 000004         2960#   7702    7708    7712
LSC      = 004000         3014#   6335    6359    8857    8875    8877
LST      = 000002         2961#   7720    7726    7730
LSTOP      060506        13886   13946#
MCLK     = 004000         2933#  12529   12560   12731   13042   13243   13362
MCPE     = 020000         3065#   6692
MDF      = 000100         2938#   7341    7389   11157   11162   11164   11517   11522   11524   11906   11911   11913   12201
                         12206   12208   12513   12518   12520   12529   12531   12549   12551   12560   12562   12581   12583
                         12715   12720   12722   12731   12733   13042   13044   13227   13232   13234   13243   13245   13346
                         13351   13353   13362   13364
MDPE     = 000400         3082#
MI       = 000004         2941#  12012   12167
MOC      = 000400         2936#   7522    7556   11229   11240   11242   11249   11254   11256   11283   11294   11296   11354
                         11383   11595   11610   11612   11619   11624   11626   11654   11669   11671   11731   11765   11988
                         12003   12005   12015   12020   12022   12028   12030   12032   12037   12039   12067   12082   12084
                         12143   12158   12160   12170   12175   12177   12188   12190   12201   12206   12208   12235   12267
                         12269   12332   12347   12349   12356   12361   12363   12371   12373   12389   12436   12451   12453
                         12458   12463   12465   12491   12506   12508   12513   12518   12520   12529   12531   12549   12551
                         12560   12562   12581   12583   12608   12623   12625   12643   12645   12672   12708   12710   12715
                         12720   12722   12731   12733   12764   12797   12799   12806   12821   12823   12907   12940   12942
                         12965   12998   13000   13042   13044   13081   13110   13112   13120   13125   13127   13164   13193
                         13195   13203   13208   13210   13243   13245   13272   13305   13307   13318   13320   13328   13330
                         13346   13351   13353   13362   13364   13454   13492   13494   13508   13539   13541   13556   13566
                         13568
MOH      = 020000         2968#
MOL      = 010000         2879#   7141    7154    7156    7179    7183
MRD      = 002000         2934#
MS       = 000040         2939#  12030
MSC      = 000002         2942#  12028   12030   12032
MSE      = 100000         3022#
MSER     = 000200         2937#   7452    7477   11322   11327   11329   11697   11702   11704   12110   12115   12117   13026
                         13031   13033
MUR      = 001000         2935#   7151    7181    8163    8216    8246    8248    8359    8401    8403    8533    8559    8561
                          8628    8657    8659    8784    8804    8806    8910    8912    8937    8939    8964    8966    9007
                          9081    9103    9105    9163    9170    9172    9222    9231    9233    9282    9300    9302    9349
                          9362    9364    9400    9415    9417    9470    9476    9478    9529    9535    9537    9636    9640
                          9642    9747    9758    9760    9845    9854    9856    9915    9921    9923   10017   10023   10025
                         10088   10105   10107   10112   10114   10152   10162   10443   10449   10451   10501   10557   10601
                         10760   10768   10770   10827   10945   10951   10953   11087   11098   11100   11139   11150   11152
                         11157   11162   11164   11189   11200   11202   11229   11240   11242   11247   11249   11254   11256
                         11283   11294   11296   11301   11306   11308   11322   11327   11329   11354   11381   11383   11438
                         11453   11455   11495   11510   11512   11517   11522   11524   11550   11565   11567   11595   11610
                         11612   11617   11619   11624   11626   11654   11669   11671   11676   11681   11683   11697   11702
                         11704   11731   11763   11765   11827   11842   11844   11884   11899   11901   11906   11911   11913
                         11942   11957   11959   11988   12003   12005   12012   12015   12020   12022   12028   12030   12032
                         12037   12039   12067   12082   12084   12089   12094   12096   12110   12115   12117   12143   12158
                         12160   12167   12170   12175   12177   12188   12190   12201   12206   12208   12235   12267   12269
                         12332   12347   12349   12354   12356   12361   12363   12371   12373   12389   12436   12451   12453
                         12491   12506   12508   12513   12518   12520   12529   12531   12549   12551   12560   12562   12581
                         12583   12608   12623   12625   12643   12645   12672   12708   12710   12715   12720   12722   12731
                         12733   12764   12797   12799   12804   12806   12821   12823   12907   12940   12942   12965   12998
                         13000   13005   13010   13012   13019   13021   13026   13031   13033   13042   13044   13081   13110
                         13112   13118   13120   13125   13127   13164   13193   13195   13201   13203   13208   13210   13215
```

```
                        13220   13222   13227   13232   13234   13243   13245   13272   13305   13307   13313   13318   13320
                        13328   13330   13346   13351   13353   13362   13364   13454   13492   13494   13506   13508   13539
                        13541   13554   13556   13566   13568   13973   14006
MWD     = 000010         2959#
MWP     = 000010         2940#   7237    7261   10871   10879   10883
MXF     = 001000         3081#
NDTMSK= 115760           2907#
NED     = 010000         3078#   5256    5271    5277    5290
NEM     = 004000         3079#
NOP     = 000000         2823#   8273    8277    8688
NSA     = 100000         2966#
OCC     = 100000         2947#   8971    9031    9239    9243
OFD     = 000200         2979#   5345    6448    6450    6518    9515    9551   13406   13522
OFFSET= 000014           2829#   8457    8712    9472   10794   14012
OM      = 000001         2886#   9458    9484    9543    9593    9645   14015   14017
ONES      066134        15119#
OPE     = 020000         3012#   6335    6359   12949
OPI     = 020000         2892#   6212    6244   11020   11044   11046   11117   11119   11208   11212   11472   11474   11573
                        11577   11861   11863   11965   11969   12395   12400   12470   12472   12947   13133   13135   15060
                        15061   15062   15063   15064   15065   15066   15067   15068   15069   15070   15071   15072   15073
                        15074   15075   15076   15077   15078   15079   15080   15081   15082   15083   15084   15085   15086
                        15087   15088   15089   15090   15091
OR      = 000200         3084#
PACACK= 000022           2833#   8906   13979   13989
PAKACK= 000022           2832#   2833    8305    8724    9129
PAR     = 000010         2902#   6196    6279    6281    6616    6635    6651
PAT     = 000020         3087#   6638    6647    6650
PCLOCK    060356        13872   13910#
PCOUNT    060444        13874   13931#
PDA     = 000400         2954#
PGE     = 002000         3080#
PGM     = 001000         2882#  10229   10312   10359   10406
PHA     = 000200         2955#
PIP     = 020000         2878#   7525    7536    7538    7554    7558
PIRQ    = 177772         2712#
PIRQVE= 000240           2806#
PLCLK     060402        13912   13916#
PLFS    = 002000         2952#
PLSTP     060512        13944   13948#
PROMPT    065135         5133    5141   14976#
PR0     = 000000         2729#
PR1     = 000040         2730#
PR2     = 000100         2731#
PR3     = 000140         2732#   4989    5200
PR4     = 000200         2733#
PR5     = 000240         2734#  13923
PR6     = 000300         2735#   6938    6979   13868   13875   13888   13921
PR7     = 000340         2736#  13920
PS      = 177776         2709#   2710
PSEL    = 002000         3067#
PSTOP     060500        13873   13943#
PSW     = 177776         2710#
PWRVEC= 000024           2801#   4960*   4961*  14875*  14876*  14885*  14891*  14903*  14904*
QSTMRK    065141         5049    5151   14977#
RD      = 000070         2853#   8317    9803   10878   12446   12501   12618   12682   12774   12917   12975   13087   13170
                        13282   13407
```

```
RDCHR = 104411        5041     5055     5065     5134     5142     14728    14866#
RDLIN = 104412        14798    14867#
RDOCT = 104413        5077     5093     5114     14868#
RDY   = 000200        3070#
READY   006212        5046     5201#    13645    13693
RECAL = 000006        2826#    8454     8700     11093    11145    11195    11235    11289    11360
RESREG= 104415        13841    14870#
RESVEC= 000010        2796#
REX   = 010000        2950#    10926    10964    10969
RG    = 040000        2948#    9693     9710     12631    12633    12689    12694    12781    12786    12924    12929    12982    12987
                      13094    13099    13177    13182    13289    13294    13467    13472
RGDTPT  066074        15102#
RH    = 000072        2854#    8325     9794
RIP   = 000020        2831#    8297     8720     8933     9126     9296     9358     9411     10882
RLEASE= 000012        2828#    8289     8589     8708     10300    10347    10394    10445
RMAS  = 000016        3031#    5304     5758*    10189    10355*   10508*   10522    10528    10598*   10613*   10653*   10693*   10718
RMASI   001344        3297#
RMASO   001420        3327#
RMBA  = 000004        3113#    5300     10096*   13451*
RMBAE = 000050        3116#    6974     6983
RMBAEI  001376        3310#    6983*    6984*
RMBAEO  001452        3340#
RMBAI   001332        3292#
RMBAO   001406        3322#    13409*   13451
RMCS1 = 000000        3027#    3111#    5339*    5350     5392*    5402     5482*    5493*    5506     5555*    5568     5820*    5996
                      6103*    6107     6112     6118*    6120*    6122     6127     6133*    6135*    6137     6143     6155*    6157*
                      6159     6165     6691     6981*    7025*    7029     7795*    7847*    7935*    8018*    8172*    8174     8239*
                      8369*    8371     8405     8544*    8546     8563     8647*    8802*    8864*    8906*    8933*    8960*    9041*
                      9098*    9167*    9226*    9296*    9358*    9411*    9472*    9531*    9638*    9689*    9752*    9849*    9917*
                      10019*   10100*   10159*   10257    10300*   10308    10347*   10394*   10445*   10560*   10724    10764*   10834*
                      10947*   11033*   11093*   11145*   11168    11195*   11235*   11289*   11313    11333    11360*   11448*   11505*
                      11528    11560*   11605*   11664*   11688    11708    11741*   11837*   11894*   11917    11952*   11998*   12077*
                      12101    12121    12153*   12212    12245*   12342*   12446*   12501*   12587    12618*   12682*   12774*   12917*
                      12975*   13087*   13170*   13282*   13460*   13979*   14012*
RMCS1I  001326        3290#    5350*    5360*    5402*    5408*    5410     5506*    5519*    5523*    5529     5568*    5580*    5585
RMCS1O  001402        3320#    13407*   13460
RMCS2 = 000010        3114#    5252*    5253*    5255     5262     5271     5273*    5274*    5276     5283     5290     5302     5335*
                      5336*    5388*    5389*    5398*    5399*    5436*    5437*    5478*    5479*    5543*    5544*    5604*    5605*
                      5726*    5727*    5749*    5750*    5781*    5782*    5813*    5814*    5837*    5838*    5869*    5870*    5892*
                      5893*    5914*    5915*    5993*    5994*    6020*    6021*    6098*    6099*    6104*    6105*    6191*    6192*
                      6273*    6274*    6321*    6322*    6403*    6404*    6442*    6443*    6564*    6565*    6610*    6611*    6630*
                      6683*    6684*    6720*    6721*    6763*    6764*    6832*    6833*    6919*    6920*    6940     7020*    7021*
                      7026*    7027*    7057*    7058*    7131*    7132*    7221*    7222*    7297*    7298*    7426*    7427*    7512*
                      7513*    7596*    7597*    7679*    7680*    7779*    7780*    7836*    7837*    7903*    7904*    7924*    7925*
                      7990*    7991*    8048*    8049*    8082*    8083*    8108*    8109*    8158*    8159*    8211*    8212*    8354*
                      8355*    8528*    8529*    8623*    8624*    8779*    8780*    8845*    8846*    9002*    9003*    9076*    9077*
                      9277*    9278*    9344*    9345*    9395*    9396*    9672*    9673*    10147*   10148*   10220*   10221*   10254*
                      10255*   10492*   10493*   10548*   10549*   10589*   10590*   10644*   10645*   10686*   10687*   10922*   10923*
                      11016*   11017*   13968*   13969*
RMCS2I  001336        3294#
RMCS2O  001412        3324#    6603*    6604*    6630     6638     6647     6650*
RMCS3 = 000052        3117#    6990
RMCS3I  001400        3311#
RMCS3O  001454        3341#
RMDA  = 000006        3028#    5341*    5352     5484*    5495*    5508     5548*    5557*    5570     5608*    5619     5752*    5764
                      5784*    5796     5872*    5880     5897*    5905     5927*    6030*    6032*    6044*    6046*    6064*    6066*
```

```
                      6725*   6729*   6733    6767*   6769*   6771    6776    6782*   6784*   6786    6791    6800*   6802*
                      6804    6809    7698*   7766    7788*   7804    7841*   7855    7929*   7995*   9289*   9307    9406*
                      9422    9426    9633*   9740*   9838*   9912*   10012*  10092*  10195   10402*  10721   10941*  11440*
                      11497*  11552*  11597*  11656*  11733*  11829*  11886*  11944*  11990*  12069*  12145*  12237*  12340*
                      12442*  12497*  12614*  12678*  12770*  12913*  12973*  13070*  13153*  13278*  13436*
RMDAI     001334      3293#   5352*   5362    5508*   5524*   5530    5570*   5586    5619*   5631    5764*   5769    5796*
                      5801    5880*   5883    5905*   5908    6733*   6736
RMDAO     001410      3323#   7825*   7841    7866*   7868*   7909*   7929    7961*   13403*  13404*  13436   13530
RMDB    = 000022      3115#   5306
RMDBI     001350      3299#
RMDBO     001424      3329#
RMDC    = 000034      3037#   5343*   5354    5501*   5514    5552*   5561*   5574    5612*   5623    5731*   5739    5818*
                      5826    5844*   5856    5899*   5919*   5931    6727*   6731*   6735    6836    6840*   6842    6850*
                      6852*   6854    6862*   6864*   6866    6881*   6883*   6887    7907    7912*   7942    7997*   9291*
                      9311    9408*   9431    9435    9467*   9590*   9742*   9840*   9908*   10014*  10094*  10943*  11442*
                      11499*  11554*  11599*  11658*  11735*  11831*  11888*  11946*  11992*  12071*  12147*  12239*  12338*
                      12444*  12499*  12616*  12680*  12772*  12915*  12971*  13072*  13155*  13280*  13438*
RMDCI     001362      3304#   5354*   5364*   5514*   5521*   5527*   5533    5574*   5582*   5588    5623*   5641    5739*
                      5742*   5743    5826*   5830*   5831    5856*   5862*   5863    5931*   5936    5937    6735*   6738*
                      6739
RMDCO     001436      3334#   13405*  13438   13488
RMDS    = 000012      3029#   5876*   5923*   7134    7140    7153    7165    7178    7219    7227    7239    7258    7516
                      7524    7535    7553    7983    7999    8024    8051    8061    8073    8093    8119    8388    8393
                      8430    8433    8916    8923    8943    8950    9014    9016    9023    9072    9087    9109    9457
                      9483    9506    9542    9576    9592    9617    9644    10225   10233   [illegible] 10271 10311   10358
                      10405   10511   10515   10562   10567   10603   10607   10617   10621   [illegible] 10663 10699   10706
                      10775   10779   11261   11266   11631   11636   12044   12049   13981   13985   14014   14019
RMDSI     001340      3295#
RMDSO     001414      3325#
RMDT    = 000026      3034#   5846*   5964    5972    10223   10456
RMDTI     001354      3301#
RMDTO     001430      3331#
RMEC1   = 000044      3041#   5850*
RMEC1I    001372      3308#
RMEC1O    001446      3338#
RMEC2   = 000046      3042#   5308    5794*   6952
RMEC2I    001374      3309#
RMEC2O    001450      3339#
RMER1   = 000014      3030#   5394*   5404    5440*   5447    5486*   5497*   5510    5563*   5576    5614*   5625    5729*
                      5737    5816*   5824    5840*   5852    5895*   5903    5925*   6190*   6194    6200    6208    6216
                      6221*   6223*   6225    6231    6239    6248    6254*   6256*   6258    6263    6276*   6278    6289
                      6613    6619    6634    6922    6927    6960    7310*   7322    7327    7359    7364    7784*   8014*
                      8057*   8069*   8087*   8091*   8113*   8165*   8218*   8361*   8535*   8630*   8786*   8852*   8904*
                      8931*   8958*   9009*   9043    9047    9083*   9176    9183    9284*   9351*   9402*   9404*   9679*
                      9765    9861    9868    9928    9933    10030   10035   10117   10121   10143   10154*  10165   10497*
                      10553*  10594*  10649*  10656*  10691*  10846   10852   10958*  11013   11019   11043   11089*  11116
                      11121   11141*  11191*  11207   11211   11231*  11285*  11356*  11444*  11471   11476   11501*  11556*
                      11572   11576   11601*  11660*  11737*  11833*  11860   11865   11890*  11948*  11964   11968   11994*
                      12073*  12149*  12241*  12334*  12394   12402   12438*  12469   12474   12493*  12610*  12674*  12766*
                      12909*  12946   12951   12967*  13083*  13132   13137   13166*  13274*  13456*  13975*  14008*
RMER1I    001342      3296#   5404*   5412    5447*   5452*   5510*   5525*   5531    5576*   5589    5625*   5647    5737*
                      5740    5824*   5827*   5828    5852*   5857    5903*   5906    6194*   6195    6203    6211    6225*
                      6226    6234    6243
RMER1O    001416      3326#   8898*   8904    8919    8931    8946    8958    8973    8982    8991*
RMER2   = 000042      3040#   5396*   5406    5442*   5449    5490*   5503*   5516    5565*   5578    5616*   5627    5756*
                      5768    5788*   5800    5848*   5874*   5882    5901*   5921*   5933    6315    6320*   6324    6351*
```

		6353*	6355	6375*	6377*	6379	6406*	6408	6615	7301	7308*	7312	7343	7347
		7382*	7386	7395	7430	7436*	7438	7450*	7454	7470*	7474	7786*	8016*	8059*
		8071*	8089*	8115*	8117*	8167*	8220*	8363*	8537*	8632*	8775	8788*	8790	8808
		8842	8854*	8856	8874	9011*	9085*	9286*	9353*	9681*	10156*	10499*	10555*	10596*
		10651*	11091*	11143*	11193*	11233*	11287*	11358*	11446*	11503*	11558*	11603*	11662*	11739*
		11835*	11892*	11950*	11996*	12075*	12151*	12243*	12336*	12440*	12495*	12612*	12676*	12768*
		12911*	12969*	13085*	13168*	13276*	13458*	13977*	14010*					
RMER2I	001370	3307#	5406*	5409*	5414	5449*	5454*	5516*	5522*	5528*	5534	5578*	5583*	5590
		5627*	5651	5768*	5774*	5775	5800*	5806*	5807	5882*	5885*	5886	5933*	5939*
		5940	6324*	6327	6334	6341	6355*	6358	6365	6379*	6381	6388	6615*	6620
RMER2O	001444	3337#												
RMHR =	000036	3038#	5735*	5792*	6024	6034	6048	6068	6632*	6686*	6689			
RMHRI	001364	3305#	6689*											
RMHRO	001440	3335#	6605*	6607*	6632	6645	6652*	6675*	6680*	6686	6699	17015		
RMLA =	000020	3032#	5790*											
RMLAI	001346	3298#												
RMLAO	001422	3328#												
RMMR1 =	000024	3033#	5444*	5451	5733*	7054	7060	7068*	7070*	7072	7081*	7083*	7085	7099*
		7101*	7103	7138*	7149*	7151*	7174*	7176*	7225*	7235*	7237*	7254*	7256*	7306*
		7339*	7341*	7380*	7384*	7434*	7448*	7452*	7468*	7472*	7520*	7522*	7533*	7549*
		7551*	7593	7599	7607*	7609*	7611*	7613	7621*	7623*	7625	7639*	7641*	7643
		7683	7700	7782*	7793*	7800*	7802*	7843*	7845*	7850*	7852*	7931*	7933*	7938*
		7940*	8012*	8020*	8022*	8055*	8085*	8111*	8161*	8163*	8214*	8216*	8246*	8248*
		8357*	8359*	8401*	8403*	8531*	8533*	8559*	8561*	8619	8626*	8628*	8635	8657*
		8659*	8661	8782*	8784*	8804*	8806*	8848*	8850*	8872*	8910*	8912*	8937*	8939*
		8964*	8966*	8970	8977	9005*	9007*	9030	9034	9079*	9081*	9103*	9105*	9163*
		9170*	9172*	9222*	9231*	9233*	9238	9248	9280*	9282*	9300*	9302*	9347*	9349*
		9362*	9364*	9398*	9400*	9415*	9417*	9470*	9476*	9478*	9529*	9535*	9537*	9636*
		9640*	9642*	9669	9675*	9677*	9692	9747*	9758*	9760*	9845*	9854*	9856*	9915*
		9921*	9923*	10017*	10023*	10025*	10088*	10105*	10107*	10112*	10114*	10150*	10152*	10192
		10443*	10449*	10451*	10495*	10501*	10551*	10557*	10592*	10601*	10615*	10647*	10689*	10760*
		10768*	10770*	10829*	10838*	10841*	10925	10929	10945*	10951*	10953*	10963	10971	11027*
		11029*	11037*	11039*	11087*	11098*	11100*	11105*	11109*	11111*	11139*	11150*	11152*	11157*
		11162*	11164*	11189*	11200*	11202*	11229*	11240*	11242*	11247*	11249*	11254*	11256*	11283*
		11294*	11296*	11301*	11306*	11308*	11322*	11327*	11329*	11354*	11381*	11383*	11438*	11453*
		11455*	11460*	11464*	11466*	11495*	11510*	11512*	11517*	11522*	11524*	11550*	11565*	11567*
		11595*	11610*	11612*	11617*	11619*	11624*	11626*	11654*	11669*	11671*	11676*	11681*	11683*
		11697*	11702*	11704*	11731*	11763*	11765*	11827*	11842*	11844*	11849*	11853*	11855*	11884*
		11899*	11901*	11906*	11911*	11913*	11942*	11957*	11959*	11988*	12003*	12005*	12012*	12015*
		12020*	12022*	12028*	12030*	12032*	12037*	12039*	12067*	12082*	12084*	12089*	12094*	12096*
		12110*	12115*	12117*	12143*	12158*	12160*	12167*	12170*	12175*	12177*	12184	12188*	12190*
		12196	12201*	12206*	12208*	12235*	12267*	12269*	12332*	12347*	12349*	12354*	12356*	12361*
		12363*	12371*	12373*	12375	12383	12389*	12436*	12451*	12453*	12458*	12463*	12465*	12491*
		12506*	12508*	12513*	12518*	12520*	12529*	12531*	12533	12541	12549*	12551*	12560*	12562*
		12568	12573	12581*	12583*	12608*	12623*	12625*	12630	12635	12643*	12645*	12672*	12688
		12696	12708*	12710*	12715*	12720*	12722*	12731*	12733*	12735	12743	12764*	12780	12788
		12797*	12799*	12804*	12806*	12821*	12823*	12907*	12923	12931	12940*	12942*	12965*	12981
		12989	12998*	13000*	13005*	13010*	13012*	13019*	13021*	13026*	13031*	13033*	13042*	13044*
		13046	13054	13081*	13093	13101	13110*	13112*	13118*	13120*	13125*	13127*	13164*	13176
		13184	13193*	13195*	13201*	13203*	13208*	13210*	13215*	13220*	13222*	13227*	13232*	13234*
		13243*	13245*	13247	13255	13272*	13288	13296	13305*	13307*	13313*	13318*	13320*	13328*
		13330*	13332	13340	13346*	13351*	13353*	13362*	13364*	13367	13375	13454*	13466	13474
		13492*	13494*	13506*	13508*	13539*	13541*	13554*	13556*	13566*	13568*	13971*	13973*	14006*
RMMR1I	001352	3300#	5451*	5456	7700*	7701	7719							
RMMR1O	001425	3330#												
RMMR2 =	000040	3039#	5762*	8207	8224	8254	10303	10315	10319	10350	10362	10366	10397	10409

```
                      10413  10458  10471  11365  11371  11746  11753  12250  12257  12649  12655  12811  12838
                      13485  13499  13513  13547  13561  13573
RMMR2I   001366        3306#
RMMR20   001442        3336#
RMOF   = 000032        3036#  5345*  5356   5488*  5499*  5512   5550*  5559*  5572   5610*  5621   5754*  5766
                       5786*  5798   5842*  5854   5878*  5917*  5929   6445*  6447   6452   6475   6480*  6482*
                       6484   6503*  6505*  6507   6532*  6534*  6536   7696*  7790*  7839*  7914*  7927*  7993*
                       9293*  9315   9340   9355*  9369   9515*  9550   9555   9631*  9744*  9842*  9910* 10090*
                      13440*
RMOFI    001360        3303#  5356*  5366*  5512*  5520*  5526*  5532   5572*  5581*  5587   5621*  5635   5766*
                       5771*  5772   5798*  5803*  5804   5854*  5859*  5860   5929*  5934*  6484*  6486   6493
                       6507*  6509   6516
RMOFO    001434        3333#  7684*  7696   7704   7716   7722   7742   7744*  7767*  7790   7791   7839   7873
                       7875*  7908*  7914   7915   7927   7958   7960* 13406* 13417* 13440  13522
RMR    = 000004        2903#  6196   6227  10166  10171  10196  10990
RMSN   = 000030        3035#  5760*  5822*  6567   6571   6575
RMSNI    001356        3302#
RMSNO    001432        3332#
RMWC   = 000002        3112#  5298  10098* 13449*
RMWCI    001330        3291#
RMWCO    001404        3321# 13408* 13449
RQA    = 100000        2990# 10304  10316  10351  10363  10398  10410  10459  10463  11366  11747  12251  12650
                      12812  13486  13514  13562
RQB    = 040000        2991# 10304  10316  10351  10363  10398  10410  10459  10465  11366  11747  12251  12650
                      12812  13486  13514  13562
RTC    = 000016        2830#  8460   8716   9531  10796
SADMSK = 000037        2872#
SAVREG = 104414       13712  14869#
SA1    = 000001        2867#
SA16   = 000020        2863#
SA2    = 000002        2866#
SA4    = 000004        2865#
SA8    = 000010        2864#
SC     = 100000        3063#
SCTMSK = 003700        2923#
SC0    = 000100        2921#
SC1    = 000200        2920#
SC2    = 000400        2919#
SC3    = 001000        2918#
SC4    = 002000        2917#
SEARCH = 000030        2836#  8313   8592   8736   9782   9917  10019  11837  11894  11952  11998  12077  12153
                      12245  12342
SEEK   = 000004        2825#  8451   8696   9785  11448  11505  11560  11605  11664  11741
SETOM    060656        9517   9584   9625  13074  13157  13442  14004#
SETVV    060526        9156   9215   9461   9509   9523   9578   9619   9733   9830   9900  10005  10080  10293
                      10340  10387  10436  10753  10820  10934  11080  11131  11182  11222  11276  11347  11430
                      11486  11542  11587  11646  11723  11819  11875  11934  11980  12059  12135  12227  12325
                      12429  12484  12601  12665  12757  12900  12958  13063  13146  13265  13429  13967#
SIZCLK   060116        5183  13864#
SKI    = 040000        3011#  5454   6409   6411   7439   7455   7457   7475   7479
SNGPRT = 020024        2971#  5965   5969
STACK  = 001100        2700#  4952   5246   5332   5385   5433   5474   5688   5960   5990   6017   6094   6184
                       6310   6439   6470   6561   6598   6670   6717   6759   6828   6911   7017   7050   7128
                       7214   7294   7423   7509   7588   7676   7762   7900   7979   8045   8150   8203   8347
                       8520   8615   8771   8838   8895   9068   9150   9207   9273   9336   9389   9453   9502
                       9572   9613   9663   9727   9824   9893   9999  10076  10139  10216  10250  10289  10336
```

```
                         10383   10431   10488   10545   10585   10640   10680   10746   10814   10916   11008   11072   11422
                         11811   12321   12423   12894   13393
STANDA   005234           5020    5031#
START    004542           3149    4940#  14569   14652
STKLMT=  177774           2711#
STOP     001530           3362#   8869    9698    9703   10968   10977   12399   12406   12693   12700   12785   12792   12928
                         12935   12986   12993   13098   13105   13181   13188   13293   13300   13471   13478   13873*  13886*
SWR      001154           3202#   4950    4972*   4974    4980*   4987*   5005    5194   13713   14333   14347   14350   14365
                         14372   14492   14499   14511   14515   14572   14611   14666*  14883   14896*
SWREG    000176           3131#   4980    5005    5194   14572   14611   14634
SW0    = 000001           2764#
SW00   = 000001           2754#   2764
SW01   = 000002           2753#   2763
SW02   = 000004           2752#   2762
SW03   = 000010           2751#   2761
SW04   = 000020           2750#   2760
SW05   = 000040           2749#   2759
SW06   = 000100           2748#   2758
SW07   = 000200           2747#   2757
SW08   = 000400           2746#   2756
SW09   = 001000           2745#   2755
SW1    = 000002           2763#
SW10   = 002000           2744#
SW11   = 004000           2743#
SW12   = 010000           2742#
SW13   = 020000           2741#  13713
SW14   = 040000           2740#
SW15   = 100000           2739#
SW2    = 000004           2762#
SW3    = 000010           2761#
SW4    = 000020           2760#
SW5    = 000040           2759#
SW6    = 000100           2758#
SW7    = 000200           2757#
SW8    = 000400           2756#
SW9    = 001000           2755#
TADMSK=  003400           2871#
TAG    = 020000           2992#  11395   11396   11397   11398   11399   11400   11784   11785   11786   11787   11788   11789
                         12291   12292   12293   12294   12295   12296   12650   12847   12848   12849   12850   12851   12852
                         13588   13589   13590   13591   13592   13593   13599   13600   13601   13602   13603   13604   13631
                         13632   13633   13634   13635
TAGADR=  001114           3174#   3181
TAP    = 040000           2967#
TA1    = 000400           2862#
TA2    = 001000           2861#
TA4    = 002000           2860#
TBITVE=  000014           2797#
TIME     001522           3359#  13916*  13933*  13935*
TKVEC  = 000060           2804#  14549*  14550*
TPVEC  = 000064           2805#
TRAPVE=  000034           2803#   4958*   4959*
TRE    = 040000           3064#
TRTVEC=  000014           2798#
TST    = 010000           2993#   8225    8227    8228    8255    8283    8291    8299    8307    8315    8323   11366   11747
                         12251   12812   13486   13514   13562
TSTNMB   060050          13724*  13725*  13727   13844#
```

```
TSTQUE  001456          3349#    5166    5167*    5203    5248    5334    5387    5435    5476    5690    5962    5992    6019
                        6096     6186    6312     6441    6472    6563    6600    6672    6719    6761    6830    6913    7019
                        7052     7130    7216     7296    7425    7511    7590    7678    7764    7902    7981    8047    8152
                        8205     8349    8522     8617    8773    8840    8897    9070    9152    9209    9275    9338    9391
                        9455     9504    9574     9615    9665    9729    9826    9895    10001   10078   10141   10218   10252
                        10291    10338   10385    10433   10490   10547   10587   10642   10682   10748   10816   10918   11010
                        11074    11424   11813    12323   12425   12896   13395   13640   13642*  13646*
TST1    006226          5237#    14392
TST10   012102          5981#    14399
TST100  037724          10374#   14455
TST101  040106          10422#   14456
TST102  040326          10479#   14457
TST103  040566          10536#   14458
TST104  040752          10576#   14459
TST105  041214          10631#   14460
TST106  041402          10672#   14461
TST107  041634          10737#   14462
TST11   012224          6008#    14400
TST110  042062          10805#   14463
TST111  042370          10895#   14464
TST112  042732          10999#   14465
TST113  043204          11063#   14466
TST114  044626          11413#   14467
TST115  046404          11802#   14468
TST116  050570          12312#   14469
TST117  051200          12414#   14470
TST12   012522          6085#    14401
TST120  053352          12885#   14471
TST121  055670          13384#   14472
TST13   013134          6175#    14402
TST14   013762          6301#    14403
TST15   014516          6430#    14404
TST16   014654          6461#    14405
TST17   015230          6552#    14406
TST2    006602          5324#    14393
TST20   015370          6589#    14407
TST21   015712          6661#    14408
TST22   016066          6708#    14409
TST23   016252          6751#    14410
TST24   016562          6819#    14411
TST25   017120          6903#    14412
TST26   017602          7008#    14413
TST27   017742          7041#    14414
TST3    007022          5376#    14394
TST30   020266          7119#    14415
TST31   020640          7205#    14416
TST32   021166          7285#    14417
TST33   021742          7414#    14418
TST34   022312          7500#    14419
TST35   022642          7579#    14420
TST36   023222          7668#    14421
TST37   023610          7753#    14422
TST4    007236          5424#    14395
TST40   024330          7891#    14423
TST41   024660          7970#    14424
TST42   025122          8036#    14425
```

```
TST43      025562        8141#   14426
TST44      025770        8194#   14427
TST45      026406        8338#   14428
TST46      027154        8511#   14429
TST47      027506        8606#   14430
TST5       007424        5466#   14396
TST50      030146        8762#   14431
TST51      030440        8829#   14432
TST52      030704        8886#   14433
TST53      031574        9059#   14434
TST54      032104        9141#   14435
TST55      032336        9198#   14436
TST56      032600        9264#   14437
TST57      033032        9327#   14438
TST6       010562        5679#   14397
TST60      033250        9380#   14439
TST61      033524        9444#   14440
TST62      033722        9493#   14441
TST63      034176        9563#   14442
TST64      034342        9604#   14443
TST65      034544        9655#   14444
TST66      035032        9718#   14445
TST67      035300        9815#   14446
TST7       011742        5951#   14398
TST70      035556        9884#   14447
TST71      036044        9990#   14448
TST72      036306       10067#   14449
TST73      036550       10130#   14450
TST74      037034       10207#   14451
TST75      037200       10241#   14452
TST76      037362       10280#   14453
TST77      037542       10327#   14454
TYPBN  = 104406        13834    14861#
TYPDS  = 104405        13674    13681    13828    14860#
TYPE   = 104401         4999     5040     5043     5049     5054     5057     5060     5064     5067     5073     5076     5082     5087
                        5092     5098     5108     5113     5119     5132     5133     5136     5141     5144     5151    13668    13675
                       13682    13717    13718    13726    13735    13741    13750    13788    13794    13798    13811    13817    13819
                       13837    13839    14086    14152    14221    14281    14494    14502    14566    14578    14632    14633    14636
                       14647    14657    14668    14687    14735    14741    14746    14750    14755    14756    14758    14761    14765
                       14856#   14905
TYPOC  = 104402         5075    13831    14635    14857#
TYPON  = 104404        14859#
TYPOS  = 104403         5089     5110    13721    13729    13738    13744    14858#
T1         006270        5243     5244     5245#
T10        012144        5987     5988     5989#
T100       037766       10380    10381    10382#
T101       040150       10428    10429    10430#
T102       040370       10485    10486    10487#
T103       040630       10542    10543    10544#
T104       041014       10582    10583    10584#
T105       041256       10637    10638    10639#
T106       041444       10677    10678    10679#
T107       041676       10743    10744    10745#
T11        012266        6014     6015     6016#
T110       042124       10811    10812    10813#
T111       042432       10913    10914    10915#
```

T112	042774	11005	11006	11007#
T113	043246	11069	11070	11071#
T114	044670	11419	11420	11421#
T115	046446	11808	11809	11810#
T116	050632	12318	12319	12320#
T117	051242	12420	12421	12422#
T12	012564	6091	6092	6093#
T120	053414	12891	12892	12893#
T121	055732	13390	13391	13392#
T13	013176	6181	6182	6183#
T14	014024	6307	6308	6309#
T15	014560	6436	6437	6438#
T16	014716	6467	6468	6469#
T17	015272	6558	6559	6560#
T2	006644	5329	5330	5331#
T20	015432	6595	6596	6597#
T21	015754	6667	6668	6669#
T22	016130	6714	6715	6716#
T23	016314	6756	6757	6758#
T24	016624	6825	6826	6827#
T25	017162	6908	6909	6910#
T26	017644	7014	7015	7016#
T27	020004	7047	7048	7049#
T3	007064	5382	5383	5384#
T30	020330	7125	7126	7127#
T31	020702	7211	7212	7213#
T32	021230	7291	7292	7293#
T33	022004	7420	7421	7422#
T34	022354	7506	7507	7508#
T35	022704	7585	7586	7587#
T36	023264	7673	7674	7675#
T37	023652	7759	7760	7761#
T4	007300	5430	5431	5432#
T40	024372	7897	7898	7899#
T41	024722	7976	7977	7978#
T42	025164	8042	8043	8044#
T43	025624	8147	8148	8149#
T44	026032	8200	8201	8202#
T45	026450	8344	8345	8346#
T46	027216	8517	8518	8519#
T47	027550	8612	8613	8614#
T5	007466	5471	5472	5473#
T50	030210	8768	8769	8770#
T51	030502	8835	8836	8837#
T52	030746	8892	8893	8894#
T53	031636	9065	9066	9067#
T54	032146	9147	9148	9149#
T55	032400	9204	9205	9206#
T56	032642	9270	9271	9272#
T57	033074	9333	9334	9335#
T6	010624	5685	5686	5687#
T60	033312	9386	9387	9388#
T61	033566	9450	9451	9452#
T62	033764	9499	9500	9501#
T63	034240	9569	9570	9571#
T64	034404	9610	9611	9612#

I13


```
T65        034606        9660     9661     9662#
T66        035074        9724     9725     9726#
T67        035342        9821     9822     9823#
T7         012004        5957     5958     5959#
T70        035620        9890     9891     9892#
T71        036106        9996     9997     9998#
T72        036350       10073    10074    10075#
T73        036612       10136    10137    10138#
T74        037076       10213    10214    10215#
T75        037242       10247    10248    10249#
T76        037424       10286    10287    10288#
T77        037604       10333    10334    10335#
UBUSQS     065177        5064    14984#
UNS      = 040000        2891#    5452     6282     6284     7317     7323     7328     7356     7360     7365     8991
UNTMSK=    000007        3095#    5259     5280     6604    10504
UPE      = 020000        3077#
USE      = 040000        3023#
U0       = 000001        3092#
U1       = 000002        3091#
U2       = 000004        3090#
VV       = 000100        2885#    7166     8917     8944     9024     9088     9110     9113    13982    13986
WATCH      001524        3360#    8865*    8867     9686*    9696    10959*   10966    12390*   12397    12684*   12691    12776*   12783
                        12919*   12926    12977*   12984    13089*   13096    13172*   13179    13284*   13291    13462*   13469    13936*
                        13938*
WC       = 000040        2957#
WCD      = 000050        2845#    8285     8960     9708     9800
WCE      = 040000        3076#   15080    15081
WCEHI    = 010000        3101#
WCELO    = 004000        3102#
WCF      = 000040        2900#    2907     6204     6235
WCH      = 000052        2846#    8293     9791
WD       = 000060        2849#    8301     9797    10100    10870    10874    10947
WH       = 000062        2850#    8309     9788
WLE      = 004000        2894#    2907    10847    10872    15084    15085
WRL      = 004000        2880#    7228     7240     7242     7259     7263
XNUDC    = 176000        2985#    5364     5521     5561     5582     5642     5645     5742     5830     5862     5936     6738     6843
                         6855     6867     6885     9312     9432
XNUER2=    001567        3018#    5409     5522     5565     5583     5652     5655     5774     5806     5885     5939     6328     6382
                         6389     6390     6416     8122
XNUOF    = 161577        2980#    5366     5520     5559     5581     5636     5639     5771     5803     5859     5934     6487     6494
                         6510     6517     6538     9316     9370
ZEROS      066176       15136#
$APTHD     001100        3161     3167#
$ASTAT=    ****** U     14948    14963
$ATYC      064710       14919    14921#
$ATY1      064664       14917#
$ATY3      064672       14266    14918#
$ATY4      064702       14507    14920#
$AUTOB     001150        3199#    5009*    5198*   14625    14782
$BASE      001276        3268#    5074     5084*    5247     5314     5333     5386     5434     5475     5689     5961     5991     6018
                         6095     6185     6311     6440     6471     6562     6599     6671     6718     6760     6829     6912     7018
                         7051     7129     7215     7295     7424     7510     7589     7677     7763     7901     7980     8046     8151
                         8204     8348     8521     8616     8772     8839     8896     9069     9151     9208     9274     9337     9390
                         9454     9503     9573     9614     9664     9728     9825     9894    10000    10077    10140    10217    10251
                        10290    10337    10384    10432    10489    10546    10586    10641    10681    10747    10815    10917    11009
                        11073    11423    11812    12322    12424    12895    13394
```

```
$BDADR  001136      3194#   5261*   5262*   5282*   5283*   5314*   5971*   5972*   6000*   6023*   6024*   6111*   6112*
                    6126*   6127*   6142*   6143*   6164*   6165*   6199*   6200*   6207*   6208*   6215*   6216*   6230*
                    6231*   6238*   6239*   6247*   6248*   6262*   6263*   6288*   6289*   6314*   6315*   6451*   6452*
                    6474*   6475*   6566*   6567*   6618*   6619*   6775*   6776*   6790*   6791*   6808*   6809*   6835*
                    6836*   6926*   6927*   6954*   7022*   7053*   7054*   7133*   7134*   7218*   7219*   7300*   7301*
                    7326*   7327*   7346*   7347*   7363*   7364*   7394*   7395*   7429*   7430*   7515*   7516*   7592*
                    7593*   7682*   7683*   7765*   7766*   7906*   7907*   7982*   7983*   8050*   8051*   8153*   8206*
                    8207*   8376*   8392*   8393*   8407*   8432*   8433*   8524*   8618*   8619*   8774*   8775*   8841*
                    8842*   8922*   8923*   8949*   8950*   8976*   8977*   9013*   9014*   9033*   9034*   9046*   9047*
                    9071*   9072*   9182*   9183*   9247*   9248*   9339*   9340*   9425*   9426*   9434*   9435*   9456*
                    9457*   9505*   9506*   9554*   9555*   9575*   9576*   9616*   9617*   9668*   9669*   9867*   9868*
                    9932*   9933*  10034*  10035*  10120*  10121*  10142*  10143*  10232*  10261*  10270*  10271*  10318*
                   10319*  10365*  10366*  10412*  10413*  10470*  10471*  10514*  10515*  10527*  10528*  10566*  10567*
                   10606*  10607*  10620*  10621*  10662*  10663*  10705*  10706*  10778*  10779*  10851*  10852*  10928*
                   10929*  10970*  10971*  11012*  11013*  11120*  11121*  11171*  11210*  11211*  11265*  11266*  11317*
                   11336*  11370*  11371*  11475*  11476*  11531*  11575*  11576*  11635*  11636*  11692*  11711*  11752*
                   11753*  11864*  11865*  11920*  11967*  11968*  12048*  12049*  12105*  12124*  12195*  12196*  12216*
                   12256*  12257*  12382*  12383*  12401*  12402*  12473*  12474*  12540*  12541*  12572*  12573*  12591*
                   12634*  12635*  12654*  12655*  12695*  12696*  12742*  12743*  12787*  12788*  12837*  12838*  12930*
                   12931*  12950*  12951*  12988*  12989*  13053*  13054*  13101*  13136*  13137*  13183*  13184*  13254*
                   13255*  13295*  13296*  13339*  13340*  13374*  13375*  13473*  13474*  13498*  13499*  13546*  13547*
                   13572*  13573*  13984*  13985*  14018*  14019*  17007   17009   17011   17017   17019   17021
$BDDAT  001142      3196#   5255*   5256    5276*   5277    5537*   5594*   5665*   5964*   5965    5967    5996*   5997*
                    6034*   6035*   6048*   6049*   6051    6068*   6069*   6070    6107*   6108*   6122*   6123*   6137*
                    6138*   6140    6159*   6160*   6161    6195*   6196*   6203*   6204*   6211*   6212*   6226*   6227*
                    6234*   6235*   6243*   6244*   6258*   6260    6278*   6281*   6284*   6285    6327*   6328*   6334*
                    6335*   6341*   6342*   6358*   6359*   6365*   6366*   6381*   6382*   6388*   6389*   6391    6408*
                    6411*   6414*   6417    6447*   6448*   6486*   6487*   6493*   6494*   6509*   6510*   6516*   6517*
                    6519    6536*   6539    6575*   6576    6613*   6616*   6634*   6635*   6636    6691*   6692*   6771*
                    6772    6786*   6787    6804*   6805    6842*   6843*   6854*   6855*   6866*   6867*   6869    6887*
                    6888    6922*   6923*   6960*   6961*   6962    7029*   7030*   7060*   7061*   7072*   7073*   7085*
                    7086*   7103*   7104*   7140*   7141*   7153*   7154*   7165*   7166*   7178*   7179*   7184    7227*
                    7228*   7239*   7240*   7258*   7259*   7264    7312*   7313*   7322*   7323*   7324    7343*   7344*
                    7359*   7360*   7361    7386*   7387*   7392    7438*   7439*   7454*   7455*   7474*   7475*   7480
                    7524*   7525*   7535*   7536*   7553*   7554*   7559    7599*   7600*   7613*   7614*   7625*   7626*
                    7643*   7644*   7647    7701*   7702*   7713    7719*   7720*   7731    7804*   7805    7813    7817
                    7855*   7856    7863    7866    7942*   7943    7953    7999*   8000*   8024*   8025*   8061*   8062*
                    8073*   8074*   8093*   8094*   8119*   8120*   8125    8174*   8175    8178*   8224*   8225    8227*
                    8254*   8255*   8257    8371*   8372    8374*   8383    8388*   8389*   8390    8405*   8406*   8412
                    8418    8425    8430*   8431*   8434    8546*   8547    8549*   8563*   8564*   8567    8573    8635*
                    8636*   8661*   8662*   8667    8790*   8791*   8808*   8809*   8812    8856*   8857*   8874*   8875*
                    8916*   8917*   8943*   8944*   8970*   8971*   9016*   9017*   9023*   9024*   9030*   9031*   9043*
                    9044*   9087*   9088*   9109*   9110*   9176*   9177*   9180    9238*   9239*   9244    9369*   9370*
                    9422*   9423    9431*   9432*   9483*   9484*   9542*   9543*   9550*   9551*   9592*   9593*   9644*
                    9645*   9692*   9693*   9694    9861*   9862*   9865    9928*   9929*  10030*  10031*  10117*  10118*
                   10165*  10166*  10168   10225*  10229*  10257*  10258*  10266*  10267*  10315*  10316*  10362*  10363*
                   10409*  10410*  10458*  10459*  10463   10465   10511*  10512*  10522*  10524*  10562*  10563*  10603*
                   10604*  10617*  10618*  10658*  10659*  10699*  10700*  10702   10775*  10776*  10846*  10847*  10848
                   10925*  10926*  10963*  10964*  11019*  11020*  11043*  11044*  11116*  11117*  11168*  11169*  11207*
                   11208*  11261*  11262*  11313*  11314*  11333*  11334*  11365*  11366*  11367   11471*  11472*  11528*
                   11529*  11572*  11573*  11631*  11632*  11688*  11689*  11708*  11709*  11746*  11747*  11750   11850*
                   11861*  11917*  11918*  11964*  11965*  12044*  12045*  12101*  12102*  12121*  12122*  12184*  12185*
                   12212*  12213*  12250*  12251*  12254   12375*  12376*  12394*  12395*  12469*  12470*  12533*  12534*
                   12568*  12569*  12587*  12588*  12630*  12631*  12649*  12650*  12652   12688*  12689*  12735*  12736*
                   12780*  12781*  12811*  12812*  12818   12923*  12924*  12946*  12947*  12981*  12982*  13046*  13047*
                   13093*  13094*  13132*  13133*  13176*  13177*  13247*  13248*  13288*  13289*  13332*  13333*  13367*
```

```
                        13368*  13466*  13467*  13485*  13486*  13489   13513*  13514*  13536   13561*  13562*  13563   13981*
                        13982*  14014*  14015*  17007   17010   17011   17013   17015   17017   17019   17021
$BELL    001212          3219#  14494   14527   14578   14775
$BIN     061126         14082*  14085*  14086   14093#
$CDW1    001302          3270#
$CDW2    001304          3271#
$CHARC   062060         14283*  14293*  14300   14309*  14314#
$CKSWR   063336         14611#  14865
$CMTAG   001114          3182#   4947    4948    4956    4962    4963    4964
$CM3   = 000000          3212#
$CM4   = 000005          3212#   3213#   3214#   3215#   3216#   3217#
$CNTLC   064234         14566   14647   14775#
$CNTLG   064246         14632   14777#
$CNTLU   064241         14657   14750   14776#
$CPUOP   001250          3242#
$CRLF    001217          3221#  13682   13717   13750   13788   13798   13811   13819   13839   14282   14317   14502   14527
                        14668   14755   14775
$DBLK    061344         14118   14152   14160#
$DDW0    001306          3272#
$DDW1    001310          3273#
$DDW2    001312          3274#
$DDW3    001314          3275#
$DDW4    001316          3276#
$DDW5    001320          3277#
$DDW6    001322          3278#
$DDW7    001324          3279#
$DEVCT   001232          3233#
$DEVM    001300          3269#   5026*   5131*   5139*   5157*   5158    5165
$DOAGN   057304         13664   13685   13691#
$DTBL    061334         14121   14156#
$ENDAD   057274          3140    4997   13687#  14522
$ENDCT   057140          4962   13666#
$ENULL   057310         13694#
$ENV     001242          3238#   5003    5019    5192   14261   14504   14926   14950
$ENVM    001243          3239#   4985    5024   14263   14268   14928
$EOP     057104         13656#
$EOPCT   057132          4962*  13663#  13667
$EOSP    057050          5316    5974   13638#
$ERFLG   001117          3185#  14322   14361   14363   14369*  14391   14489*  14527
$ERMAX   001131          3191#   4965*   5242*   5328*   5381*   5429*   5470*   5684*   5956*   5986*   6013*   6090*   6180*
                         6306*   6435*   6466*   6557*   6594*   6666*   6713*   6755*   6824*   6907*   7013*   7046*   7124*
                         7210*   7290*   7419*   7505*   7584*   7672*   7758*   7896*   7975*   8041*   8146*   8199*   8343*
                         8516*   8611*   8767*   8834*   8891*   9064*   9146*   9203*   9269*   9332*   9385*   9449*   9498*
                         9568*   9609*   9659*   9723*   9820*   9889*   9995*  10072*  10135*  10212*  10246*  10285*  10332*
                        10379*  10427*  10484*  10541*  10581*  10636*  10676*  10742*  10810*  10912*  11004*  11068*  11418*
                        11807*  12317*  12419*  12890*  13389*  14363   14386*  14391
$ERROR   062640          4956   14487#
$ERRPC   001132          3192#  13742   14496*  14497*  14498   14527
$ERRTB   001532          3380#  13757
$ERTTL   001126          3189#  13679   13683*  14495*  14527
$ESCAP   001210          3218#   4964*  14385*  14518   14520   14527
$ETABL   001242          3237#
$ETEND   001326          3173    3280#
$FATAL   001224          3230#  14954*
$FFLG    065130         14917*  14920*  14948   14957*  14965#
$FILLC   001172          3210#  14286   14317
```

```
$FILLS  001171     3209#   14317
$GDADR  001134     3193#   10233*
$GDDAT  001140     3195#   5258*   5259*   5260*   5279*   5280*   5281*   5538*   5595*   5664*   5999*   6037*   6050*
                   6051    6072*   6110*   6125*   6139*   6140    6163*   6198*   6206*   6214*   6229*   6237*   6246*
                   6259*   6260    6287*   6316*   6390*   6391    6415*   6416*   6417    6450*   6476*   6518*   6519
                   6541*   6571*   6576    6602*   6636    6651*   6676*   6774*   6789*   6807*   6834*   6868*   6869
                   6890*   6914*   6925*   6962    6965    6986*   6992*   7032*   7063*   7075*   7088*   7108*   7143*
                   7156*   7168*   7187*   7220*   7242*   7266*   7302*   7317*   7324    7328    7337*   7350*   7354
                   7356*   7361    7365    7388*   7391*   7392    7441*   7457*   7476*   7479*   7480    7527*   7538*
                   7555*   7558*   7559    7594*   7628*   7649*   7703*   7708*   7712*   7713    7721*   7726*   7730*
                   7731    7769*   7805    7812*   7815*   7816*   7826*   7856    7862*   7865*   7877*   7910*   7943
                   7949*   7950    7952*   8002*   8027*   8064*   8066*   8121*   8124*   8125    8179*   8228*   8256*
                   8257    8375*   8382*   8385*   8390    8408*   8420*   8424*   8427*   843[illegible]   8542*   8575*   8638*
                   8663*   8666*   8667    8793*   8810*   8811*   8812    8859*   8877*   8921*   8948*   8975*   9012*
                   9090*   9113*   9178*   9179*   9180    9240*   9243*   9244    9341*   9392*   9458*   9545*   9553*
                   9595*   9647*   9666*   9694    9710*   9863*   9864*   9865    9931*   10033*  10122*  10167*  10168
                   10231*  10260*  10269*  10516*  10526*  10565*  10608*  10622*  10661*  10701*  10702   10780*  10850*
                   10930*  10969*  11022*  11046*  11119*  11172*  11212*  11264*  11316*  11337*  11369*  11474*  11532*
                   11577*  11634*  11691*  11712*  11748*  11749*  11750   11863*  11921*  11969*  12047*  12104*  12125*
                   12194*  12215*  12252*  12253*  12254   12381*  12400*  12472*  12539*  12571*  12590*  12633*  12651*
                   12652   12694*  12741*  12786*  12813*  12817*  12818   12929*  12949*  12987*  13052*  13099*  13100*
                   13135*  13182*  13253*  13294*  13338*  13373*  13472*  13487*  13488*  13489   13515*  13521*  13524*
                   13525*  13533*  13536   13563   13986*  14017*  17007   17010   17013   17015   17017   17019
$GET42  057264     13684#
$GTSWR  063426     14633#  14863
$HD   = 000001     2677    2678
$HIBTS  001100     3168#
$HIOCT  064376     14815*  14820#
$ICNT   001120     3186#   5241*   5327*   5380*   5428*   5469*   5683*   5955*   5985*   6012*   6089*   6179*   6305*
                   6434*   6465*   6556*   6593*   6665*   6712*   6754*   6823*   6906*   7012*   7045*   7123*   7209*
                   7289*   7418*   7504*   7583*   7671*   7757*   7895*   7974*   8040*   8145*   8198*   8342*   8515*
                   8610*   8766*   8833*   8890*   9063*   9145*   9202*   9268*   9331*   9384*   9448*   9497*   9567*
                   9608*   9658*   9722*   9819*   9888*   9994*   10071*  10134*  10211*  10245*  10284*  10331*  10378*
                   10426*  10483*  10540*  10580*  10635*  10675*  10741*  10809*  10911*  11003*  11067*  11417*  11806*
                   12316*  12418*  12889*  13388*  14376*  14377   14379*  14390
$ILLUP  064646     14875   14891   14908#
$INTAG  001151     3200#   14630*  14649   14669   14782
$ITEMB  001130     3190#   13733   14498*  14506   14527
$LF     001220     3222#   14317   14527   14765   14775
$LFLG   065127     14958*  14964#
$LLCSR  001512     3355#   13884   13914*  13946*
$LLVEC  001514     3356#   13876*  13877*  13887*  13888*  13899*  13900*
$LPADR  001122     3187#   4966*   5243*   5329*   5382*   5430*   5471*   5685*   5957*   5987*   6014*   6091*   6181*
                   6307*   6436*   6467*   6558*   6595*   6667*   6714*   6756*   6825*   6908*   7014*   7047*   7125*
                   7211*   7291*   7420*   7506*   7585*   7673*   7759*   7897*   7976*   8042*   8147*   8200*   8344*
                   8517*   8612*   8768*   8835*   8892*   9065*   9147*   9204*   9270*   9333*   9386*   9450*   9499*
                   9569*   9610*   9660*   9724*   9821*   9890*   9996*   10073*  10136*  10213*  10247*  10286*  10333*
                   10380*  10428*  10485*  10542*  10582*  10637*  10677*  10743*  10811*  10913*  11005*  11069*  11419*
                   11808*  12318*  12420*  12891*  13390*  14358*  14367*  14383*  14388   14390
$LPCSB  001504     3352#   13910*
$LPCSR  001502     3351#   13871   13911*  13943*
$LPERR  001124     3188#   4967*   5244*   5330*   5383*   5431*   5472*   5686*   5958*   5988*   6015*   6092*   6182*
                   6308*   6437*   6468*   6559*   6596*   6668*   6715*   6757*   6826*   6909*   7015*   7048*   7126*
                   7212*   7292*   7421*   7507*   7586*   7674*   7760*   7898*   7977*   8043*   8148*   8201*   8345*
                   8518*   8613*   8769*   8836*   8893*   9066*   9148*   9205*   9271*   9334*   9387*   9451*   9500*
                   9570*   9611*   9661*   9725*   9822*   9891*   9997*   10074*  10137*  10214*  10248*  10287*  10334*
```

```
                       10381*  10429*  10486*  10543*  10583*  10638*  10678*  10744*  10812*  10914*  11006*  11070*  11124*
                       11175*  11215*  11269*  11340*  11420*  11479*  11535*  11580*  11639*  11715*  11809*  11868*  11927*
                       11972*  12052*  12128*  12219*  12319*  12421*  12477*  12594*  12658*  12750*  12892*  12953*  13056*
                       13139*  13258*  13391*  14367   14384*  14390   14517
$LPVEC   001506         3353#  13874*  13875*  13889*  13890*  13897*  13898*
$MADR1   001254         3255#
$MADR2   001260         3259#
$MADR3   001264         3262#
$MADR4   001270         3265#
$MAIL    001222         3169    3173    3228#   4984    5003    5192    5238    5325    5377    5425    5467    5680    5952
                        5982    6009    6086    6176    6302    6431    6462    6553    6590    6662    6709    6752    6820
                        6904    7009    7042    7120    7206    7286    7415    7501    7580    7669    7754    7892    7971
                        8037    8142    8195    8339    8512    8607    8763    8830    8887    9060    9142    9199    9265
                        9328    9381    9445    9494    9564    9605    9656    9719    9816    9885    9991   10068   10131
                       10208   10242   10281   10328   10375   10423   10480   10537   10577   10632   10673   10738   10806
                       10896   11000   11064   11414   11803   12313   12415   12886   13385   14261   14382   14504
$MAMS1   001252         3249#
$MAMS2   001256         3257#
$MAMS3   001262         3260#
$MAMS4   001266         3263#
$MBADR   001102         3169#
$MFLG    065126        14918*  14924   14959*  14963#
$MNEW    064264        14636   14780#
$MSGAD   001236         3235#  14934*  14937
$MSGLG   001240         3236#  14939*
$MSGTY   001222         3229#  14932   14940*  14952   14956*
$MSWR    064253        14633   14778#
$MTYP1   001253         3250#
$MTYP2   001257         3258#
$MTYP3   001263         3261#
$MTYP4   001267         3264#
$MXCNT   062374        14380   14390#
$NULL    001170         3208#  14288   14317
$NWTST=  000001         5233#   5235    5320#   5322    5372#   5374    5420#   5422    5462#   5464    5675#   5677    5947#
                        5949    5977#   5979    6004#   6006    6081#   6083    6171#   6173    6297#   6299    6426#   6428
                        6457#   6459    6548#   6550    6585#   6587    6657#   6659    6704#   6706    6747#   6749    6815#
                        6817    6899#   6901    7004#   7006    7037#   7039    7115#   7117    7201#   7203    7281#   7283
                        7410#   7412    7496#   7498    7575#   7577    7664#   7666    7749#   7751    7887#   7889    7966#
                        7968    8032#   8034    8137#   8139    8190#   8192    8334#   8336    8507#   8509    8602#   8604
                        8758#   8760    8825#   8827    8882#   8884    9055#   9057    9137#   9139    9194#   9196    9260#
                        9262    9323#   9325    9376#   9378    9440#   9442    9489#   9491    9559#   9561    9600#   9602
                        9651#   9653    9714#   9716    9811#   9813    9880#   9882    9986#   9988   10063#  10065   10126#
                       10128   10203#  10205   10237#  10239   10276#  10278   10323#  10325   10370#  10372   10418#  10420
                       10475#  10477   10532#  10534   10572#  10574   10627#  10629   10668#  10670   10733#  10735   10801#
                       10803   10891#  10893   10995#  10997   11059#  11061   11409#  11411   11798#  11800   12308#  12310
                       12410#  12412   12881#  12883   13380#  13382
$OCNT    061576        14193*  14222*  14235#
$OMODE   061600        14188*  14192*  14197   14200*  14211*  14237#
$OVER    062360        14334   14359   14368   14378   14387#
$PASS    001230         3232#   4984*  13660*  13661*  13672   13694   14374   14391
$PASTM   001106         3171#
$POWER   064654        14906   14911#
$PSW     001520         3358#  13919*  13949
$PWRDN   064506         4960   14875#  14903
$PWRMG   064642        14906#
$PWRUP   064560        14885   14891#
```

```
$QUES    001216      3220#   14317   14527   14687   14758   14775
$RDCHR   063700     14700#   14866
$RDDEC= ****** U    14869
$RDLIN   063770     14723#   14867
$RDOCT   064276     14793#   14868
$RDSZ  = 000010     14716#
$RESRE   061016     14058#   14870
$RTNAD   057306     13693#
$R2A   = ****** U   14871
$SAVRE   060760     14042#   14869
$SAVR6   064652     14884*   14892   14893*  14894*  14910#
$SCOPE   062064      4954    14331#
$SETUP=  000137      3122#    4953    4954    4956    4958    4960    4962    4963    4964    4966    4997    5000    5189
                    13658    14332   14488   14514   14522   14570   14575   14576   14606   14782
$STUP  = 177777      3122#
$SVLAD   062324     14342    14381#
$SVPC  = 000200      3138#    3143
$SWR   = 167400      2666#    2677    2682    2683    2684    2685    2686    2687    2688    3217    3218    3219    4963
                     4964     4966    4967    5238    5325    5377    5425    5467    5680    5952    5982    6009    6086
                     6176     6302    6431    6462    6553    6590    6662    6709    6752    6820    6904    7009    7042
                     7120     7206    7286    7415    7501    7580    7669    7754    7892    7971    8037    8142    8195
                     8339     8512    8607    8763    8830    8887    9060    9142    9199    9265    9328    9381    9445
                     9494     9564    9605    9656    9719    9816    9885    9991   10068   10131   10208   10242   10281
                    10328    10375   10423   10480   10537   10577   10632   10673   10738   10806   10896   11000   11064
                    11414    11803   12313   12415   12886   13385   13653   13659   13686   13692   13694   14323   14324
                    14325    14326   14327   14333   14345   14347   14348   14361   14362   14363   14370   14371   14372
                    14384    14387   14390   14479   14480   14481   14482   14483   14492   14499   14511   14515   14527
                    14907
$SWREG   001244      3240#    4987
$SWRMK=  000000      2688     2689   14327   14328   14351
$SWOBT   062376     14357    14391#
$TESTN   001226      3231#    5238*   5325*   5377*   5425*   5467*   5680*   5952*   5982*   6009*   6086*   6176*   6302*
                     6431*    6462*   6553*   6590*   6662*   6709*   6752*   6820*   6904*   7009*   7042*   7120*   7206*
                     7286*    7415*   7501*   7580*   7669*   7754*   7892*   7971*   8037*   8142*   8195*   8339*   8512*
                     8607*    8763*   8830*   8887*   9060*   9142*   9199*   9265*   9328*   9381*   9445*   9494*   9564*
                     9605*    9656*   9719*   9816*   9885*   9991*  10068*  10131*  10208*  10242*  10281*  10328*  10375*
                    10423*   10480*  10537*  10577*  10632*  10673*  10738*  10806*  10896*  11000*  11064*  11414*  11803*
                    12313*   12415*  12886*  13385*  13725   14382*
$TIMES   001206      3217#    4963*  13659*  14370*  14377   14380*  14390
$TKB     001162      3205#   14530   14551   14562   14587   14615   14642
$TKCNT   063040     14531#   14546*  14576   14593*  14707   14709*
$TKINT   063050      5033     5202   14546#  14567   14628
$TKQEN=  063047     14535#   14601   14712
$TKQIN   063042     14532#   14547*  14548   14599*  14600*  14601   14603*
$TKQOU   063044     14533#   14548*  14710   14711*  14712   14714*
$TKQSR   063046     14534#   14547   14603   14714
$TKS     001160      3204#   14530   14552*  14583*  14585   14591*  14613   14629*  14639   14651*  14671*
$TKSRV   063120     14549    14562#
$TMP0    001174      3212#    5969*   6964*   7106*   7107*   7186*   7267*   7396*   7482*   7561*   7650*   7715*   7733*
                     7791*    7915*   8096*   8127*   8240*   8648*   8814*   9112*   9184*   9246*   9699*   9869*   9934*
                    10036*   10170*  10223*  10227   10456*  10461   10704*  10781*  10853*  10972*  11047*  13989*  17011
                    17013    17017   17019
$TMP1    001176      3213#    5042*   5043    5044    5047    5056*   5057    5058    5066*   5067    5068    5078*   5080
                     5084     5085*   5086*   5088    5094*   5096    5100    5101*   5102*   5103*   5104*   5105*   5106*
                     5107*    5109    5115*   5117    5122*   5123*   5124*   5125*   5126*   5127    5135*   5136    5137
                     5143*    5144    5145    5147    5149    5153    5970*   7716*   8097*   8128*  10854*  17011   17019
```

```
$TMP2     001200     3214#
$TMP3     001202     3215#
$TMP4     001204     3216#
$TN     = 000122     2677#   5233    5238#   5243    5320    5325#   5329    5372    5377#   5382    5420    5425#   5430
                     5462    5467#   5471    5675    5680#   5685    5947    5952#   5957    5977    5982#   5987    6004
                     6009#   6014    6081    6086#   6091    6171    6176#   6181    6297    6302#   6307    6426    6431#
                     6436    6457    6462#   6467    6548    6553#   6558    6585    6590#   6595    6657    6662#   6667
                     6704    6709#   6714    6747    6752#   6756    6815    6820#   6825    6899    6904#   6908    7004
                     7009#   7014    7037    7042#   7047    7115    7120#   7125    7201    7206#   7211    7281    7286#
                     7291    7410    7415#   7420    7496    7501#   7506    7575    7580#   7585    7664    7669#   7673
                     7749    7754#   7759    7887    7892#   7897    7966    7971#   7976    8032    8037#   8042    8137
                     8142#   8147    8190    8195#   8200    8334    8339#   8344    8507    8512#   8517    8602    8607#
                     8612    8758    8763#   8768    8825    8830#   8835    8882    8887#   8892    9055    9060#   9065
                     9137    9142#   9147    9194    9199#   9204    9260    9265#   9270    9323    9328#   9333    9376
                     9381#   9386    9440    9445#   9450    9489    9494#   9499    9559    9564#   9569    9600    9605#
                     9610    9651    9656#   9660    9714    9719#   9724    9811    9816#   9821    9880    9885#   9890
                     9986    9991#   9996    10063   10068#  10073   10126   10131#  10136   10203   10208#  10213   10237
                     10242#  10247   10276   10281#  10286   10323   10328#  10333   10370   10375#  10380   10418   10423#
                     10428   10475   10480#  10485   10532   10537#  10542   10572   10577#  10582   10627   10632#  10637
                     10668   10673#  10677   10733   10738#  10743   10801   10806#  10811   10891   10896#  10913   10995
                     11000#  11005   11059   11064#  11069   11409   11414#  11419   11798   11803#  11808   12308   12313#
                     12318   12410   12415#  12420   12881   12886#  12891   13380   13385#  13390   14352   14392
$TPB      001166     3207#   14306*  14317
$TPFLG    001173     3211#   14255   14317
$TPS      001164     3206#   14304   14317
$TRAP     064400     4958    14829#
$TRAP2    064440     14844#  14855
$TRP    = 000016     14848#  14857#  14858#  14859#  14860#  14861#  14862#  14863   14864#  14865   14866#  14867#  14868#
                     14869#  14870#  14871#
$TRPAD    064452     14838   14855#
$TSTM     001104     3170#
$TSTNM    001116     3184#   13658*  14322   14354*  14381*  14382   14387   14391   14491   14527
$TTYIN    064224     14725   14726   14738   14756   14770   14774#
$TYPBN    061054     14079#  14861
$TYPDS    061130     14106#  14860
$TYPE     061602     14255#  14848   14856   14945
$TYPEC    062014     14285   14292   14299   14304#  14305   14673
$TYPEX    062062     14310   14312   14315#
$TYPOC    061400     14191#  14857
$TYPON    061414     14190   14193#  14859
$TYPOS    061354     14186#  14858
$UNIT     001234     3234#   5203*   13719
$UNITM    001110     3172#
$USWR     001246     3241#
$VECT1    001272     3266#   5085    5100*   5101    5127*
$VECT2    001274     3267#
$XTSTR    062076     14336#
$$GET4= 000000       13686#
$$SW08= 000122       14391#  14392   14393#  14394#  14395#  14396#  14397#  14398#  14399#  14400#  14401#  14402#  14403#
                     14404#  14405#  14406#  14407#  14408#  14409#  14410#  14411#  14412#  14413#  14414#  14415#  14416#
                     14417#  14418#  14419#  14420#  14421#  14422#  14423#  14424#  14425#  14426#  14427#  14428#  14429#
                     14430#  14431#  14432#  14433#  14434#  14435#  14436#  14437#  14438#  14439#  14440#  14441#  14442#
                     14443#  14444#  14445#  14446#  14447#  14448#  14449#  14450#  14451#  14452#  14453#  14454#  14455#
                     14456#  14457#  14458#  14459#  14460#  14461#  14462#  14463#  14464#  14465#  14466#  14467#  14468#
                     14469#  14470#  14471#  14472#  14473#
$0FILL    061577     14187*  14191*  14201   14236#
```

```
$4OCAT= ****** U        14333   14501
.      = 113370          3125#   3129#   3138    3139#   3141#   3143#   3148#   3151#   3157    3158#   3160#   3162#   3174
                         3181#   3223    3349#   4951    4966    4967   13671#  13694   13695#  14160#  14317   14390   14391
                        14527   14530   14534#  14535   14536#  14774#  14775   14782#  14887   14909   14966#  15056#  17031#
                        17033#  17034#  17035#
.$ASTA= ****** U        14918   14921
.$X    = 001100          3157#   3162
```

CROSS REFERENCE TABLE -- MACRO NAMES

CLEAR	2666#	5252	5273	5335	5388	5398	5436	5478	5543	5603	5726	5749	5781	5813	5837
	5869	5892	5914	5993	6020	6098	6104	6191	6272	6321	6403	6442	6564	6610	6683
	6720	6763	6832	6919	7020	7026	7057	7131	7221	7297	7426	7512	7596	7679	7779
	7836	7903	7924	7990	8048	8082	8108	8158	8210	8353	8528	8623	8779	8845	9002
	9076	9277	9344	9395	9672	10147	10220	10254	10492	10548	10589	10644	10685	10922	11016
	13968														
CLKOFF	2666#	8869	9698	9703	10968	10977	12399	12405	12693	12699	12785	12791	12928	12934	12986
	12992	13098	13104	13181	13187	13293	13299	13471	13477						
CLKON	2666#	8865	9686	10959	12390	12684	12776	12919	12977	13089	13172	13284	13462		
COMMEN	2807#														
ENDCOM	2807#														
ERR	2666#	3385	3393	3401	3409	3417	3425	3434	3442	3450	3458	3466	3474	3482	3490
	3498	3506	3514	3522	3530	3538	3546	3554	3562	3570	3578	3586	3594	3602	3610
	3618	3626	3634	3642	3650	3658	3666	3674	3682	3690	3698	3706	3714	3722	3730
	3738	3746	3754	3762	3770	3779	3786	3794	3802	3810	3818	3826	3834	3842	3850
	3858	3866	3874	3882	3890	3898	3906	3914	3922	3930	3938	3946	3954	3962	3970
	3978	3986	3994	4002	4010	4018	4026	4034	4042	4050	4058	4066	4074	4082	4090
	4098	4106	4114	4122	4130	4138	4146	4154	4162	4170	4178	4186	4194	4202	4210
	4218	4226	4234	4242	4250	4258	4266	4274	4282	4290	4298	4306	4314	4322	4330
	4338	4346	4354	4362	4370	4378	4386	4394	4402	4410	4418	4426	4434	4442	4450
	4458	4466	4474	4482	4490	4498	4506	4514	4522	4530	4538	4546	4554	4562	4570
	4578	4586	4594	4602	4611	4619	4627	4635	4643	4651	4659	4667	4675	4684	4692
	4700	4708	4716	4725	4733	4742	4750	4758	4766	4774	4782	4790	4799	4807	4815
	4823	4831	4839	4847	4855	4863	4871	4879	4887	4895	4903	4912	4920	4928	
ERROR	2701#	5185	5263	5284	5315	5369	5417	5458	5539	5596	5666	5745	5777	5809	5833
	5865	5888	5910	5942	5973	6001	6038	6053	6073	6113	6128	6144	6166	6201	6209
	6217	6232	6240	6249	6264	6290	6330	6337	6344	6361	6368	6384	6393	6419	6453
	6489	6496	6512	6521	6542	6578	6622	6624	6640	6642	6694	6741	6777	6792	6810
	6845	6857	6871	6891	6928	6955	6967	6969	7033	7064	7076	7089	7109	7144	7157
	7169	7188	7230	7243	7268	7315	7330	7333	7348	7367	7370	7397	7442	7483	7528
	7539	7562	7602	7616	7629	7651	7717	7734	7807	7858	7945	8003	8028	8065	8076
	8098	8129	8180	8229	8259	8377	8394	8414	8421	8436	8550	8569	8576	8639	8669
	8794	8815	8860	8878	8924	8951	8978	8985	9019	9026	9035	9048	9091	9114	9158
	9185	9217	9249	9319	9372	9427	9436	9463	9486	9511	9519	9525	9546	9556	9580
	9586	9596	9621	9627	9648	9700	9735	9775	9832	9870	9902	9935	10007	10037	10082
	10123	10173	10175	10234	10262	10272	10295	10320	10342	10367	10389	10414	10438	10472	10517
	10529	10568	10609	10623	10664	10707	10755	10782	10822	10855	10931	10936	10973	11023	11048
	11082	11122	11133	11173	11184	11213	11224	11267	11278	11318	11338	11349	11372	11432	11477
	11488	11533	11544	11578	11589	11637	11648	11693	11713	11725	11754	11821	11866	11877	11922
	11936	11970	11982	12050	12061	12106	12126	12137	12197	12217	12229	12258	12327	12384	12403
	12431	12475	12486	12542	12574	12592	12603	12636	12656	12667	12697	12744	12759	12789	12839
	12902	12932	12952	12960	12990	13055	13065	13076	13102	13138	13148	13159	13185	13256	13267
	13297	13341	13376	13431	13444	13475	13500	13548	13574						
ESCAPE	2807#														
GETAS	2666#	10521													
GETBA	2666#														
GETBAE	2666#	6982													
GETCS1	2666#	5348	5401	5505	5567	5995	6106	6121	6136	6158	6690	7028	8173	8370	8404
	8545	8562	10256	10307	11167	11312	11332	11527	11687	11707	11916	12100	12120	12211	12586
GETCS2	2666#	5254	5275												
GETDA	2666#	5351	5507	5569	5618	5763	5795	5879	5904	6732	6770	6785	6803	7803	7854
	9306	9421													
GETDB	2666#														
GETDC	2666#	5353	5513	5573	5622	5738	5825	5855	5930	6734	6841	6853	6865	6886	7941
	9310	9430													
GETDS	2666#	7139	7152	7164	7177	7226	7238	7257	7523	7534	7552	7998	8023	8060	8072

	8092	8118	8386	8428	8915	8942	9015	9021	9086	9108	9482	9541	9591	9643	10224
	10265	10310	10357	10404	10510	10561	10602	10616	10657	10698	10774	11260	11630	12043	13980
	14013														
GETDT	2666#	5963	10222	10455											
GETEC1	2666#														
GETEC2	2666#														
GETER1	2666#	5403	5446	5509	5575	5624	5736	5823	5851	5902	6193	6224	6257	6277	6612
	6633	6921	6958	7321	7358	9042	9175	9764	9860	9927	10029	10116	10164	10845	11018
	11042	11115	11206	11470	11571	11859	11963	12392	12468	12945	13131				
GETER2	2666#	5405	5448	5515	5577	5626	5767	5799	5881	5932	6323	6354	6378	6407	6614
	7311	7342	7385	7437	7453	7473	8789	8807	8855	8873					
GETHR	2666#	6033	6047	6067	6688										
GETLA	2666#														
GETMR1	2666#	5450	7059	7071	7084	7102	7598	7612	7624	7642	7699	8634	8660	8969	9028
	9237	9690	10924	10961	12183	12374	12532	12567	12629	12686	12734	12778	12921	12979	13045
	13091	13174	13246	13286	13331	13366	13464								
GETMR2	2666#	8223	8253	10302	10314	10349	10361	10396	10408	10457	11363	11744	12248	12648	12809
	13483	13511	13559												
GETOF	2666#	5355	5511	5571	5620	5765	5797	5853	5928	6446	6483	6506	6535	9314	9368
	9549														
GETPRI	2807#	13917													
GETSN	2666#	6570	6574												
GETSWR	2666#	2807#	5000#	5189											
GETWC	2666#														
GETX	2666#														
MSG	2666#	5235	5322	5374	5422	5464	5677	5949	5979	6006	6083	6173	6299	6428	6459
	6550	6587	6659	6706	6749	6817	6901	7006	7039	7117	7203	7283	7412	7498	7577
	7666	7751	7889	7968	8034	8139	8192	8336	8509	8604	8760	8827	8884	9057	9139
	9196	9262	9325	9378	9442	9491	9561	9602	9653	9716	9813	9882	9988	10065	10128
	10205	10239	10278	10325	10372	10420	10477	10534	10574	10629	10670	10735	10803	10893	10997
	11061	11411	11800	12310	12412	12883	13382								
MULT	2807#														
NEWTST	2807#	5233	5320	5372	5420	5462	5675	5947	5977	6004	6081	6171	6297	6426	6457
	6548	6585	6657	6704	6747	6815	6899	7004	7037	7115	7201	7281	7410	7496	7575
	7664	7749	7887	7966	8032	8137	8190	8334	8507	8602	8758	8825	8882	9055	9137
	9194	9260	9323	9376	9440	9489	9559	9600	9651	9714	9811	9880	9986	10063	10126
	10203	10237	10276	10323	10370	10418	10475	10532	10572	10627	10668	10733	10801	10891	10995
	11059	11409	11798	12308	12410	12881	13380								
POP	2807#	6944	6945	6949	6951	6987	6989	6999	7001	13903	13905	14063	14147	14816	14896
	14897	14960	14961												
PUSH	2807#	6935	6936	6976	6977	13865	13866	14043	14106	14795	14877	14883	14921	14923	14944
PUTAS	2666#	5757	10354	10507	10597	10612	10652	10692							
PUTBA	2666#	10095	13450												
PUTBAE	2666#														
PUTCS1	2666#	5338	5391	5481	5492	5554	5819	6102	6117	6119	6132	6134	6154	6156	6980
	7024	7794	7846	7934	8017	8171	8238	8368	8543	8646	8801	8862	8905	8932	8959
	9040	9097	9166	9225	9295	9357	9410	9471	9530	9637	9688	9751	9848	9916	10018
	10099	10158	10299	10346	10393	10444	10559	10763	10833	10946	11032	11092	11144	11194	11234
	11288	11359	11447	11504	11559	11604	11663	11740	11836	11893	11951	11997	12076	12152	12244
	12341	12445	12500	12617	12681	12773	12916	12974	13086	13169	13281	13459	13978	14011	
PUTCS2	2666#	6629													
PUTDA	2666#	5340	5483	5494	5547	5556	5607	5751	5783	5871	5896	5926	6029	6031	6043
	6045	6063	6065	6724	6728	6766	6768	6780	6783	6798	6801	7697	7787	7840	7928
	7994	9288	9405	9632	9739	9837	9911	10011	10091	10401	10940	11439	11496	11551	11596
	11655	11732	11828	11885	11943	11989	12068	12144	12236	12339	12441	12496	12613	12677	12769
	12912	12972	13069	13152	13277	13435									

```
PUTDB    2666#
PUTDC    2666#   5342   5500   5551   5560   5611   5730   5817   5843   5898   5918   6726   6730   6839   6849
         6851    6861   6863   6880   6882   7911   7996   9290   9407   9465   9589   9741   9839   9907  10013
        10093   10942  11441  11498  11553  11598  11657  11734  11830  11887  11945  11991  12070  12146  12238
        12337   12443  12498  12615  12679  12771  12914  12970  13071  13154  13279  13437
PUTDS    2666#   5875   5922
PUTDT    2666#   5845
PUTEC1   2666#   5849
PUTEC2   2666#   5793
PUTER1   2666#   5393   5439   5485   5496   5562   5613   5728   5815   5839   5894   5924   6189   6220   6222
         6253    6255   6275   7309   7783   8013   8056   8068   8086   8090   8112   8164   8217   8360   8534
         8629    8785   8851   8903   8930   8957   9008   9082   9283   9350   9401   9403   9678  10153  10496
        10552   10593  10648  10655  10690  10957  11088  11140  11190  11230  11284  11355  11443  11500  11555
        11600   11659  11736  11832  11889  11947  11993  12072  12148  12240  12333  12437  12492  12609  12673
        12765   12908  12966  13082  13165  13273  13455  13974  14007
PUTER2   2666#   5395   5441   5489   5502   5564   5615   5755   5787   5847   5873   5900   5920   6319   6350
         6352    6374   6376   6405   7307   7381   7435   7449   7469   7785   8015   8058   8070   8088   8114
         8116    8166   8219   8362   8536   8631   8787   8853   9010   9084   9285   9352   9680  10155  10498
        10554   10595  10650  11090  11142  11192  11232  11286  11357  11445  11502  11557  11602  11661  11738
        11834   11891  11949  11995  12074  12150  12242  12335  12439  12494  12611  12675  12767  12910  12968
        13084   13167  13275  13457  13976  14009
PUTHR    2666#   5734   5791   6631   6685
PUTLA    2666#   5789
PUTMR1   2666#   5443   5732   7067   7069   7080   7082   7097   7100   7137   7148   7150   7172   7175   7224
         7234    7236   7253   7255   7305   7338   7340   7378   7383   7433   7447   7451   7467   7471   7519
         7521    7532   7548   7550   7606   7608   7610   7620   7622   7638   7640   7781   7792   7799   7801
         7842    7844   7849   7851   7930   7932   7937   7939   8011   8019   8021   8054   8084   8110   8160
         8162    8213   8215   8245   8247   8356   8358   8400   8402   8530   8532   8558   8560   8625   8627
         8656    8658   8781   8783   8803   8805   8847   8849   8871   8908   8911   8935   8938   8962   8965
         9004    9006   9078   9080   9101   9104   9162   9168   9171   9221   9229   9232   9279   9281   9299
         9301    9346   9348   9360   9363   9397   9399   9413   9416   9469   9474   9477   9527   9533   9536
         9635    9639   9641   9674   9676   9746   9757   9759   9844   9852   9855   9914   9920   9922  10016
        10021   10024  10087  10103  10106  10111  10113  10149  10151  10442  10447  10450  10494  10500  10550
        10556   10591  10600  10614  10646  10688  10759  10767  10769  10828  10837  10840  10944  10949  10952
        11026   11028  11036  11038  11086  11096  11099  11104  11107  11110  11138  11149  11151  11156  11160
        11163   11188  11199  11201  11228  11238  11241  11246  11248  11253  11255  11282  11293  11295  11300
        11305   11307  11320  11325  11328  11353  11380  11382  11437  11451  11454  11459  11462  11465  11494
        11509   11511  11516  11520  11523  11549  11564  11566  11594  11608  11611  11616  11618  11623  11625
        11653   11668  11670  11675  11680  11682  11695  11700  11703  11730  11762  11764  11826  11840  11843
        11848   11851  11854  11883  11898  11900  11905  11909  11912  11941  11956  11958  11987  12001  12004
        12011   12014  12019  12021  12027  12029  12031  12035  12038  12066  12081  12083  12088  12093  12095
        12108   12113  12116  12142  12156  12159  12166  12169  12174  12176  12187  12189  12200  12204  12207
        12234   12266  12268  12331  12345  12348  12353  12355  12359  12362  12369  12372  12388  12435  12449
        12452   12457  12461  12464  12490  12504  12507  12512  12516  12519  12527  12530  12548  12550  12558
        12561   12579  12582  12607  12621  12624  12641  12644  12671  12706  12709  12714  12718  12721  12729
        12732   12763  12795  12798  12803  12805  12820  12822  12906  12938  12941  12964  12996  12999  13004
        13008   13011  13017  13020  13025  13029  13032  13040  13043  13080  13108  13111  13117  13119  13123
        13126   13162  13191  13194  13200  13202  13206  13209  13214  13218  13221  13226  13230  13233  13241
        13244   13271  13303  13306  13312  13316  13319  13326  13329  13345  13349  13352  13360  13363  13453
        13491   13493  13505  13507  13538  13540  13553  13555  13565  13567  13970  13972  14005
PUTMR2   2666#   5761
PUTOF    2666#   5344   5487   5498   5549   5558   5609   5753   5785   5841   5877   5916   6444   6479   6481
         6502    6504   6531   6533   7695   7789   7838   7913   7926   7992   9292   9354   9513   9630   9743
         9841    9909  10089  13439
PUTSN    2666#   5759   5821
PUTWC    2666#  10097  13448
```

PUTX	2666#														
REPORT	2807#														
RGBFMC	2666#	3283	3313												
SCOPE	2702#	5237	5324	5376	5424	5466	5679	5951	5981	6008	6085	6175	6301	6430	6461
	6552	6589	6661	6708	6751	6819	6903	7008	7041	7119	7205	7285	7414	7500	7579
	7668	7753	7891	7970	8036	8141	8194	8338	8511	8606	8762	8829	8886	9059	9141
	9198	9264	9327	9380	9444	9493	9563	9604	9655	9718	9815	9884	9990	10067	10130
	10207	10241	10280	10327	10374	10422	10479	10536	10576	10631	10672	10737	10805	10895	10999
	11063	11413	11802	12312	12414	12885	13384								
SETOM	2666#	9516	9583	9624	13073	13156	13441								
SETPRI	2807#	4989	13923	13949	14703										
SETTRA	14848#	14857	14858	14859	14860	14861	14863	14865	14866	14867	14868	14869	14870		
SETUP	2807#	4946													
SETVV	2666#	9155	9214	9460	9508	9522	9577	9618	9732	9829	9899	10004	10079	10292	10339
	10386	10435	10752	10819	10933	11079	11130	11181	11221	11275	11346	11429	11485	11541	11586
	11645	11722	11818	11874	11933	11979	12058	12134	12226	12324	12428	12482	12599	12663	12755
	12899	12957	13062	13144	13263	13427									
SKIP	2807#														
SLASH	2807#														
SPACE	2807#														
STARS	2807#	3136	3154	3156	3163	3177	3223	3226	5233	5236	5320	5323	5372	5375	5420
	5423	5462	5465	5675	5678	5947	5950	5977	5980	6004	6007	6041	6056	6081	6084
	6171	6174	6297	6300	6317	6348	6372	6397	6426	6429	6457	6460	6477	6500	6525
	6548	6551	6585	6588	6657	6660	6704	6707	6747	6750	6815	6818	6899	6902	6917
	6931	7004	7007	7037	7040	7115	7118	7201	7204	7281	7284	7410	7413	7496	7499
	7575	7578	7664	7667	7749	7752	7770	7822	7971	7887	7890	7966	7969	8032	8035
	8137	8140	8190	8193	8334	8337	8446	8507	8510	8602	8605	8758	8761	8825	8828
	8882	8885	9055	9058	9137	9140	9194	9197	9260	9263	9323	9326	9376	9379	9440
	9443	9489	9492	9559	9562	9600	9603	9651	9654	9714	9717	9811	9814	9880	9883
	9986	9989	10063	10066	10126	10129	10203	10206	10237	10240	10276	10279	10323	10326	10370
	10373	10418	10421	10475	10478	10532	10535	10572	10575	10627	10630	10668	10671	10733	10736
	10801	10804	10891	10894	10995	10998	11059	11062	11076	11126	11177	11217	11271	11342	11409
	11412	11426	11481	11537	11582	11641	11718	11798	11801	11815	11870	11929	11974	12054	12130
	12222	12308	12311	12410	12413	12426	12479	12596	12660	12752	12881	12884	12897	12954	13058
	13141	13260	13380	13383	13396	13399	13414	13424	13649	13698	13863	13908	14027	14072	14096
	14163	14240	14319	14475	14529	14606	14621	14692	14716	14785	14823	14873	14889	14916	
SWRSU	2807#	4968#													
TAGS	2666#	3282													
TRMTRP	14848#														
TSTSET	2666#	5240	5326	5379	5427	5468	5682	5954	5984	6011	6088	6178	6304	6433	6464
	6555	6592	6664	6711	6753	6822	6905	7011	7044	7122	7208	7288	7417	7503	7582
	7670	7756	7894	7973	8039	8144	8197	8341	8514	8609	8765	8832	8889	9062	9144
	9201	9267	9330	9383	9447	9496	9566	9607	9657	9721	9818	9887	9993	10070	10133
	10210	10244	10283	10330	10377	10425	10482	10539	10579	10634	10674	10740	10808	10910	11002
	11066	11416	11805	12315	12417	12888	13387								
TYPBIN	2807#														
TYPDEC	2807#	13672	13679												
TYPNAM	2666#	2807#	4993												
TYPNUM	2807#														
TYPOCS	2807#	5088	5109	13719	13727	13736	13742								
TYPOCT	2807#	5074	14634												
TYPTXT	2807#	13668	13675												
$$CMRE	3175#														
$$CMTM	3175#	3212	3213	3214	3215	3216									
$$ESCA	2807#														
$$NEWT	2807#	5233	5320	5372	5420	5462	5675	5947	5977	6004	6081	6171	6297	6426	6457

```
         6548    6585    6657    6704    6747    6815    6899    7004    7037    7115    7201    7281    7410    7496    7575
         7664    7749    7887    7966    8032    8137    8190    8334    8507    8602    8758    8825    8882    9055    9137
         9194    9260    9323    9376    9440    9489    9559    9600    9651    9714    9811    9880    9986   10063   10126
        10203   10237   10276   10323   10370   10418   10475   10532   10572   10627   10668   10733   10801   10891   10995
        11059   11409   11798   12308   12410   12881   13380
$$SET   14848#  14857   14858   14859   14860   14861   14863   14865   14866   14867   14868   14869   14870
$$SETM   4984#
$$SKIP   2807#
.EQUAT   2666#   2697
.GETPR   2666#
.HEADE   2666#   2667
.SETUP   2666#   3122
.SWRHI   2666#   2678
.SWRLO   2666#   2689#
.$ACT1   2666#   3134
.$APTB   2666#   3224#
.$APTH   2666#   3152
.$APTY   2666#  14914
.$CATC   2666#   3123
.$CMTA   2666#   3175
.$DIV    2666#
.$EOP    2666#  13647
.$ERRO   2666#  14473
.$ERRT   2666#
.$MULT   2666#
.$POWE   2666#  14871
.$RAND   2666#
.$RDDE   2666#
.$RDOC   2666#  14783
.$READ   2666#  14527
.$SAVE   2666#  14025
.$SCOP   2666#  14317
.$SIZE   2666#
.$TRAP   2666#  14821
.$TYPB   2666#  14070
.$TYPD   2666#  14094
.$TYPE   2666#  14238
.$TYPO   2666#  14161

. ABS.  113370     000

 ERRORS DETECTED:  0

 DSKZ:DZRMJA.BIN,DSKZ:DZRMJA.SEQ/DOC/SOL/CRF=DSKM:DZRMJA.P11
 RUN-TIME: 49 47 5 SECONDS
 RUN-TIME RATIO: 576/102=5.6
 CORE USED:  33K  (65 PAGES)

 DOCUMENT PAGES:  386
```