

RABO-**

UDA & DISK DRV DIAG
CZUDCAO

AH-S831A-MC
FICHE 1 OF 2

OCT 1981
COPYRIGHT © 76-81
MADE IN USA



RABO-**

UDA & DISK DRV DIAG
CZUDCAO

AH-S831A-MC
FICHE 2 OF 2

OCT 1981
COPYRIGHT © 76-81
MADE IN USA



.REM 8

IDENTIFICATION

PRODUCT CODE: AC-S830A-MC
PRODUCT NAME: CZUDCAO UDA AND DISK DRIVE DIAGNOSTIC
PRODUCT DATE: 10-JULY-81
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: DALE KECK

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

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

COPYRIGHT (C) 1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	TEST SUMMARIES

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE UDA HOST RESIDENT DIAGNOSTIC CAUSES THE EXECUTION OF THE UDA AND DRIVE RESIDENT DIAGNOSTIC INDIVIDUAL PROGRAMS AND REPORTS THE RESULTS ON THE PDP-11 CONSOLE TERMINAL OR LINE PRINTER. EXERCISES ALSO PERFORMS TO VERIFY THAT:

1. THE UDA CAN PROPERLY COMMUNICATE WITH THE PDP-11 PROCESSOR AND CAN TRANSFER BLOCKS OF DATA TO AND FROM UNIBUS MEMORY.
2. THE UDA AND THE DISK DRIVES CAN COMMUNICATE AND TRANSFER DATA PROPERLY.
3. THE DISK DRIVES CAN FUNCTION PROPERLY AS DEFINED BY THE SDI.
4. THE DISK DRIVES CAN SEEK, READ AND WRITE TO ALL BLOCKS ON THE DISK AND ALLOW THE MEASUREMENT OF THE DRIVES' PERFORMANCE IN A RATIO OF ERRORS PER MILLION BITS TRANSFERRED.

THE UDA HOST RESIDENT DIAGNOSTIC CONSISTS OF ONE PDP-11 DIAGNOSTIC SUPERVISOR PROGRAM THAT RUNS IN THE PDP-11 PROCESSOR AND FOUR PROGRAMS THAT RUNS IN THE UDA'S BUFFER MEMORY THROUGH AN INTER-PRATER CALLED THE 'DIAGNOSTIC MACHINE' WHICH RESIDES IN THE UDA. THE PDP-11 PROGRAM MAINLY IS RESPONSIBLE FOR DOWNLINE LOADING THE 'DIAGNOSTIC MACHINE' PROGRAMS INTO THE UDA AND STARTING THEIR EXECUTION. THE 'DIAGNOSTIC MACHINE' PROGRAM CONTROLS THE TESTING FROM THAT POINT BY REQUESTING THE PDP-11 PROCESSOR TO SUPPLY INFORMATION, PRINT ERROR MESSAGES AND UPDATE STATISTICS. THE 'DIAGNOSTIC MACHINE' PROGRAM INFORMS THE PDP-11 PROCESSOR WHEN A TEST IS COMPLETE.

THE NUMBER OF UDAS AND DISK DRIVES THAT CAN BE TESTED AT ONE TIME BY THIS DIAGNOSTIC PROGRAM IS LIMITED BY THE PDP-11 MEMORY SIZE. WITH THE MINIMUM MEMORY SIZE OF 28K WORDS, AT LEAST TWO UDAS WITH FOUR DISK DRIVES EACH ARE SELECTABLE. THE NUMBER OF UDAS AND DISK DRIVES SELECTABLE BY THE DIAGNOSTIC INCREASES WITH ADDITIONAL MEMORY, BUT THERE IS NO GOAL IMPLIED FOR THE MAXIMUM POSSIBLE CONFIGURATION.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

THIS PROGRAM WILL BE DESIGNED USING THE PDP-11 SUPERVISOR REVISION C. RUN TIME ENVIRONMENTS ARE DETERMINED BY THE SUPERVISOR AND MAY CHANGE AS NEW VERSIONS OF THE SUPERVISOR ARE DEVELOPED. THE INITIAL VERSION WILL REQUIRE THE FOLLOWING:

PDP-11 PROCESSOR
28K WORDS OF MEMORY (MINIMUM)
XXDP+ LOAD MEDIA
ONE OR MORE UDA SUB-SYSTEMS
LINE CLOCK - EITHER TYPE L OR P

THE LINE CLOCK WILL BE USED FOR ALL TIMED LOOPS IN THE PROGRAM. THE DIAGNOSTIC WILL RUN ON A SYSTEM WITH NO CLOCK BUT WILL HANG WHENEVER AN EVENT FOR WHICH THE PROGRAM IS WAITING DOES NOT HAPPEN (I.E., A TIME-OUT ERROR MESSAGE WILL NOT RESULT).

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ^C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.

/PASS:DDDDD EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
 /FLAGS:FLGS SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED
 IN SECTION 2.3.
 /EOP:DDDDD REPORT END OF PASS MESSAGE AFTER EVERY
 DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
 /UNITS:LIST TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED
 IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12
 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)

IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	'BELL' ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A 'BELL' ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING 'CHANGE HW (L) ?' YOU MUST ANSWER 'Y' AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN 'PRELOADED' USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A 'Y', THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

THE PROGRAM WILL ASK THE FOLLOWING QUESTIONS IN RESPONSE TO A START COMMAND (NON-SCRIPT):

CHANGE HW?

ANSWER NO TO USE THE PRE-BUILT ANSWERS FOR ALL HARDWARE QUESTIONS. THIS PROGRAM WILL BE RELEASED PRE-BUILT TO TEST ONE UNIT WITH THE DEFAULT ANSWERS SHOWN BELOW. THE PRE-BUILT ANSWERS MAY BE CHANGED AT ANY TIME WITH THE SETUP UTILITY. ANSWER 'YES' TO BE ASKED ALL THE HARDWARE QUESTIONS.

UNITS (D) ?

ANSWER WITH THE NUMBER OF UNITS TO BE TESTED (NO DEFAULT). THIS ANSWER WILL DETERMINE HOW MANY TIMES THE FOLLOWING QUESTIONS ARE ASKED. A UNIT IS A LOGICAL DISK DRIVE ON A UDA. 1 TO 64 UNITS MAY BE SPECIFIED.

UNIBUS ADDRESS OF UDA (O) 172150 ?

ANSWER WITH THE ADDRESS OF THE UDAIP REGISTER OF ONE UDA AS ADDRESSED BY THE PROCESSOR WITH MEMORY MANAGEMENT TURNED OFF (I.E., AN EVEN 16-BIT ADDRESS IN THE RANGE OF 160000 TO 177774).

VECTOR (O) 154 ?

ANSWER WITH THE INTERRUPT VECTOR ADDRESS OF THE UDA. A VECTOR ADDRESS IN THE RANGE OF 4 TO 774 MAY BE SPECIFIED.

BR LEVEL (D) 5 ?

ANSWER WITH THE INTERRUPT PRIORITY USED BY THE UDA. LEVELS 4 TO 7 ARE ACCEPTED.

UNIBUS BURST RATE (D) 0 ?

THE UDA ALLOWS THE ABILITY TO CONTROL THE MAXIMUM NUMBER OF WORDS TRANSFERRED ACROSS THE UNIBUS EACH TIME THE UDA BECOMES MASTER. ANSWER WITH THE VALUE YOUR OPERATING SYSTEM USES OR USE ZERO WHICH WILL TELL THE UDA TO SUPPLY A VALUE. A DECIMAL NUMBER IN THE RANGE OF 0 TO 63 MAY BE SPECIFIED. THE VALUE WILL BE PASSED DIRECTLY TO THE UDA DURING INITIALIZATION.

DRIVE NUMBER (D) 0 ?

ANSWER WITH THE LOGICAL DRIVE NUMBER ON THE FRONT OF THE DRIVE YOU WISH TO TEST. ON A MULTI-UNIT DRIVE, EACH SUB-UNIT NUMBER ON THE DRIVE MUST BE TESTED AS A SEPARATE UNIT TO COMPLETELY TEST THE DRIVE. A MAXIMUM OF EIGHT LOGICAL DRIVES MAY BE TESTED ON ONE UDA AT A TIME (UDA CONFIGURATION LIMIT).

EXERCISE ON CUSTOMER DATA AREA IN TEST 4 (L) N ?

ANSWER NO TO HAVE TEST 4 (DRIVE EXERCISER) RUN ON THE DIAGNOSTIC AREA OF THE DISK. ANSWER YES TO RUN ON THE CUSTOMER DATA AREA. A YES ANSWER WILL DESTROY ANY CUSTOMER DATA THAT MAY BE ON THE DISK. A WARNING MESSAGE WILL BE PRINTED IF THIS QUESTION IS ANSWERED YES (SEE SECTION 4.3).

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

THE PROGRAM WILL ASK THE FOLLOWING QUESTIONS IN RESPONSE TO THE START,

RESTART, AND CONTINUE COMMANDS.

CHANGE SW ?

ANSWER NO TO BYPASS THE FOLLOWING QUESTIONS IN THIS SECTION.
A YES ANSWER WILL CAUSE THE QUESTIONS TO BE ASKED AND ALLOW
THE DEFAULT ANSWERS TO BE CHANGED.

ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS (L) N ?

TESTS 2 AND 4 HAVE MANUAL INTERVENTION MODES WHICH ALLOW
ADDITIONAL PARAMETERS TO BE INPUT TO ALTER THE NORMAL TESTING
OF A DISK DRIVE. THIS QUESTION SHOULD NORMALLY BE ANSWERED
NO WHEN THIS DIAGNOSTIC IS FIRST RUN. THEN, DEPENDING ON THE
ERRORS DETECTED, IT MAY BE DESIRABLE TO CHANGE THIS ANSWER TO
YES AND ALTER THE TESTING TO FURTHER ISOLATE THE PROBLEM. IF
THIS QUESTION IS ANSWERED YES, AND THE UAM (UNATTENDED MODE
OPERATION) FLAG IS SET, TESTS 2 AND 4 WILL PRINT A WARNING
MESSAGE THAT THE MODE CANNOT BE ENTERED AND WILL PROCEED AS
IF ANSWERED NO. SEE THE DESCRIPTION OF THE INDIVIDUAL TESTS
FOR MORE INFORMATION.

REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY

THIS INFORMATIONAL MESSAGE IS PRINTED TO DESCRIBE THE USE OF
THE REMAINING QUESTIONS. IF TEST 4 IS NOT BEING RUN, A
"CONTROL Z" CAN BE TYPED TO BYPASS THEM.

ERROR LIMIT (D) 32 ?

ENTER THE NUMBER OF HARD ERRORS ALLOWED BEFORE A DRIVE IS
DROPPED FROM EXERCISE BY TEST #4. A NUMBER IN THE RANGE OF 1
TO 65535 WILL BE ACCEPTED.

READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT (D) 0 ?

WHEN THE SPECIFIED NUMBER OF BYTES HAVE BEEN READ FROM A
DRIVE BY TEST #4, THE DRIVE WILL BE DROPPED FROM TESTING.
WHEN ALL DRIVES ARE DROPPED, AN END OF PASS WILL BE INDICATED
AND THE SELECTED TESTS WILL BE RUN AGAIN. THIS IS THE METHOD
USED TO DETERMINE HOW LONG TEST #4 IS TO RUN. ANSWER WITH A
ZERO TO PREVENT TEST FROM ENDING. THE ONLY OTHER WAY TEST #4
CAN END IS TO HAVE ALL DRIVES DROPPED BECAUSE THE ERROR LIMIT
ON EACH IS EXCEEDED. OF COURSE, THE OPERATOR CAN ALWAYS STOP
TEST #4 BY TYPING A CONTROL-C.

SUPPRESS PRINTING SOFT ERRORS (L) Y?

WHEN TEST #4 NEEDS TO PERFORM RETRIES, SOFT ERROR REPORTS
WILL BE PRINTED TO GIVE AS MUCH INFORMATION AS POSSIBLE.
THESE ACTIONS ARE CONSIDERED NORMAL OPERATION AND ARE NOT
ERROR CONDITIONS UNTIL THE RETRIES FAIL. WHEN THE TEST IS
BEING RUN ONLY TO SEE HOW RELIABLE THE DRIVE PERFORMS, THIS
QUESTION SHOULD BE ANSWERED YES SO THEY ARE NOT CONFUSED WITH
HARD ERRORS. THE NUMBER OF THESE SOFT ERRORS IS ALWAYS
REPORTED IN THE STATISTICAL REPORT. ANSWER NO TO SEE ALL THE
SOFT ERROR REPORTS.

DO INITIAL WRITE ON START (L) Y ?

IF TEST #4 IS TO DO DATA COMPARES, THE DRIVE WILL NEED TO BE WRITTEN WITH DATA PATTERNS READABLE BY THE PROGRAM.

IF THE DIAGNOSTIC AREA IS SELECTED FOR TESTING, THE INITIAL WRITE IS ALWAYS PERFORMED (REGARDLESS OF HOW THIS QUESTION IS ANSWERED).

IF THE CUSTOMER DATA AREA IS SELECTED FOR TESTING, THE INITIAL WRITE WILL BE PERFORMED WHEN ALL OF THE FOLLOWING ARE TRUE:

1. THIS QUESTION IS ANSWERED YES.
2. THIS IS THE FIRST TIME TEST #4 IS BEING RUN AFTER A START COMMAND.
3. THE DISK IS WRITE ENABLED.

ANSWERING THIS QUESTION NO WHEN TESTING ON THE CUSTOMER DATA AREA WILL NORMALLY RESULT IN DATA COMPARISON ERRORS IF THE DISK WAS NOT PREVIOUSLY WRITTEN BY THIS DIAGNOSTIC OR THE FORMATTER.

NOTE THAT WRITE CHECKS ARE NOT PERFORMED DURING THE INITIAL WRITE.

ENABLE ERROR LOG (L) N ?

A YES ANSWER WILL CAUSE ERROR MESSAGES IN TEST #4 TO BE STORED IN A LOG BUFFER. ONCE THE LOG BUFFER IS FULL, ADDITIONAL ERROR INFORMATION IS LOST. THE CONTENTS OF THE LOG BUFFER WILL BE PRINTED WHEN TEST #4 IS STOPPED AND A STATISTICAL REPORT REQUESTED. THIS LOG FEATURE IS INTENDED TO ALLOW THE DIGITAL DIAGNOSIS CENTER (DDC) TO START TEST #4 THEN HANG UP FROM THE SYSTEM AND LET IT RUN FOR SOME PERIOD OF TIME. DDC CAN CALL THE SYSTEM BACK LATER, TYPE CONTROL-C, THEN PRINT AND SEE THE ERRORS THAT HAVE OCCURRED (UP TO THE LIMIT OF THE LOG BUFFER). A MESSAGE WILL BE PRINTED TO INDICATE NO ERRORS HAVE OCCURRED IF THE LOG BUFFER IS EMPTY. TEST #4 WILL NOT BE ALLOWED TO END WHILE THE ERROR LOG IS ENABLED EVEN THOUGH TESTING MAY ACTUALLY STOP. THE LOG BUFFER WILL HOLD A MINIMUM OF 30 ERROR MESSAGES.

2.6 EXTENDED P-TABLE DIALOGUE

THE FOLLOWING WARNING WILL BE PRINTED IN RESPONSE TO A START COMMAND IF ANY DRIVE IS SELECTED FOR TESTING ON THE CUSTOMER DATA AREA. THE WARNING WILL APPEAR IMMEDIATELY AFTER THE SOFTWARE QUESTIONING IS COMPLETED.

CUSTOMER DATA WILL BE DESTROYED ON:

UNIT UDA AT DRIVE
XX XXXXXX XXX

UNLESS THE DIAGNOSTIC IS BEING RUN IN UNATTENDED MODE (E.G.,
START/FLAG:UAM TO SUPERVISOR PROMPT), A CONFIRMATION WILL ALSO BE
REQUIRED AS FOLLOWS:

ARE YOU SURE CUSTOMER DATA CAN BE DESTROYED (L) ?

IF THE ABOVE QUESTION IS ANSWERED NO, THE ENTIRE DIAGNOSTIC WILL STOP
AND THE SUPERVISOR PROMPT WILL BE DISPLAYED. NO DEFAULT ANSWER IS
PROVIDED FOR THIS QUESTION.

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE
IS A CLOCK) QUESTIONS
3. TYPE 'R NAME', WHERE NAME IS THE NAME OF THE BIN OR BIC
FILE FOR THIS PROGRAM
4. TYPE 'START'
5. ANSWER THE 'CHANGE HW' QUESTION WITH 'Y'
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE 'CHANGE SW' QUESTION WITH 'N'

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE
DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS
ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY
A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES
ARE ALWAYS PRINTED UNLESS THE 'IER' FLAG IS SET (SECTION 2.3).
THE GENERAL ERROR MESSAGE IS OF THE FORM:

NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXXXX
ERROR MESSAGE

WHERE: NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

4.0 PERFORMANCE AND PROGRESS REPORTS

A STATISTICAL REPORT WILL AUTOMATICALLY BE PRINTED PERIODICALLY (APPROXIMATELY EVERY SEVENTEEN MINUTES) AND AT THE END OF TEST #4. IT CAN BE SUPPRESSED BY SETTING THE INHIBIT STATISTICAL REPORT FLAG (E.G. START/FLAGS:ISR). THIS IS THE SAME REPORT THAT CAN BE PRINTED ON DEMAND WITH THE PRINT COMMAND.

DURING TEST 1, 2, AND 3, THE REPORT WILL LOOK LIKE THE FOLLOWING EXAMPLE:

TEST 1 IN PROGRESS RUN TIME 2:24:10

DURING TEST #4, THE REPORT WILL CONTAIN STATISTICS ON EACH DRIVE FOR THE CURRENT PASS OF THE TEST; FOR EXAMPLE:

TEST 4 IN PROGRESS RUN TIME 2:24:10

UNIT	DRIVE	SERIAL-NUMBER	SEKS	MBYTES READ	MBYTES WRITTEN	HARD ERRORS	SOFT ERRORS	ECC
0	0	1002	12000	36	22	0	0	1
1	4	67342102112	14000	42	29	0	2	0

5.0 TEST SUMMARIES

TEST # 1 - UNIBUSS ADDRESSING TEST

THE PURPOSE OF TEST #1 IS TO COMPLETE THE TESTING OF THE UNIBUS INTERFACE IN THE UDA. THE UDA RESIDENT DIAGNOSTIC IS NOT ABLE TO COMPLETELY TEST THE UNIBUS INTERFACE BECAUSE COMMUNICATION WITH THE PDP-11 PROCESSOR IS NECESSARY. SPECIFICALLY, THIS TEST WILL:

1. CHECK THAT EVERY ADDRESS LINE ON THE UNIBUS CAN BE DRIVEN TO BOTH ONE AND ZERO STATES.
2. CHECK THAT THE UDA CAN INTERRUPT THE PDP-11 PROCESSOR AT THE PROPER PRIORITY LEVEL AND VECTOR.
3. EXERCISE THE UNIBUS INTERFACE BY TRANSFERRING BLOCKS OF DATA TO AND FROM UNIBUS MEMORY.

THIS TEST ASSUMES THAT THE FOLLOWING ARE BEING TESTED BY THE UDA RESIDENT DIAGNOSTIC:

1. ALL DATA BITS CAN BE WRITTEN AND READ CORRECTLY.
2. NPR CYCLES CAN BE EXECUTED CORRECTLY.

ONE AT A TIME, EACH UDA SELECTED FOR TESTING WILL BE INITIALIZED AND BROUGHT ON-LINE BY FOLLOWING THE INITIALIZATION PROTOCOL. SEVERAL SIZES OF THE HOST COMMUNICATIONS AREA WILL BE SUPPLIED TO ALLOW THE UDA RESIDENT DIAGNOSTIC TO DO THE MOST UNIBUS ADDRESS TESTING POSSIBLE. INTERRUPTS WILL BE DISABLED. ANY UDA RESIDENT DIAGNOSTIC ERRORS WILL BE REPORTED.

THE UDA WILL THEN BE INITIALIZED AGAIN, THIS TIME WITH INTERRUPTS ENABLED. THE VECTOR ADDRESS AND PRIORITY LEVEL WILL BE DETERMINED SOLELY FROM THE ANSWERS TO THE HARDWARE QUESTIONS. IF THE HARDWARE VECTORS TO THE WRONG ADDRESS, IT IS IMPOSSIBLE TO DETERMINE THE RESULT. A DESCRIPTIVE ERROR MESSAGE OF THE PROBLEM WILL NOT OCCUR (THE PROGRAM OR PROCESSOR MAY HANG OR AN UNRELATED MESSAGE MAY OCCUR). THEREFORE, THE MESSAGE 'TESTING INTERRUPT ABILITY OF UDA AT ADR XXXXXX VEC XXX...' WILL BE PRINTED JUST BEFORE THE UDA IS REQUESTED TO CAUSE AN INTERRUPT AND THE WORD 'COMPLETED' WILL BE PRINTED (ON THE SAME LINE) WHEN THE INTERRUPT TEST IS COMPLETED. IF THE WORD 'COMPLETED' DOES NOT FOLLOW THE FIRST MESSAGE, IT SHOULD BE APPARENT THAT THE INTERRUPT CAUSED THE DIAGNOSTIC OR PROCESSOR TO GO ASTRAY. THE PRIORITY LEVEL OF THE INTERRUPT REQUEST WILL ALSO BE VERIFIED.

A 'DIAGNOSTIC MACHINE' PROGRAM WILL THEN BE DOWNLINE LOADED IN THE UDA FROM THE MEMORY SPACE INCLUDED IN THE HOST COMMUNICATIONS AREA WHEN THE UDA WAS FIRST INITIALIZED. THE UDA RESIDENT DIAGNOSTIC HAS ALREADY VERIFIED THAT IT CAN ACCESS THESE MEMORY ADDRESSES, SO THE DOWNLINE LOAD COMMAND SHOULD PERFORM PROPERLY. THE 'DIAGNOSTIC MACHINE' PROGRAM IS THEN STARTED.

THE 'DIAGNOSTIC MACHINE' PROGRAM WILL ASK THE PDP-11 PROGRAM TO FILL FREE MEMORY (THAT MEMORY AVAILABLE TO THE PDP-11 PROGRAM THAT IS NOT BEING USED BY THE PROGRAM OR THE PDP-11 SUPERVISOR) WITH AN ADDRESSING PATTERN AND REPORT THE LOCATION AND SIZE OF THE FREE MEMORY. EVERY LOCATION OF FREE MEMORY WILL BE READ AND THE DATA CHECKED. THEN, ONE BY ONE, EACH ADDRESS LINE WILL BE TESTED AS FOLLOWS:

1. DETERMINE A TEST ADDRESS BY TAKING THE FIRST ADDRESS OF FREE MEMORY AND COMPLEMENTING THE ADDRESS BIT TO BE TESTED.
2. READ FROM THE TEST ADDRESS.
3. IF A NON-EXISTANT MEMORY ERROR OCCURRS, THE TEST IS COMPLETE.
4. WRITE ALL ONES TO THE FIRST ADDRESS OF FREE MEMORY THEN READ FROM THE TEST ADDRESS. IF DATA READ IS NOT ALL ONES, THEN TEST IS COMPLETE.
5. WRITE ZEROS TO THE FIRST ADDRESS OF FREE MEMORY THEN READ FROM THE TEST ADDRESS. IF DATA READ IS NOT ZEROS, THEN TEST IS COMPLETE.
6. REPORT UNIBUS ADDRESSING ERROR.

WHEN ALL ADDRESS BITS HAVE BEEN TESTED, THEN BLOCK TRANSFERS TO AND

FROM MEMORY WILL BE TESTED WITH DIFFERENT DATA PATTERNS. THIS DATA WILL BE TRANSFERRED AT THE RATE DISK DATA IS TRANSFERRED TO AND FROM MEMORY DURING NORMAL UDA OPERATION.

THE NEXT UDA SELECTED FOR TESTING WILL THEN BE TESTED IN THE SAME MANNER. WHEN ALL UDAS HAVE BEEN TESTED, TEST #1 WILL END.

TEST #2 - DISK RESIDENT DIAGNOSTIC TEST

THE PURPOSE OF TEST #2 IS TO EXECUTE THE DIAGNOSTICS THAT RUN IN EACH DISK DRIVE. THESE DIAGNOSTIC PROGRAMS MAY BE RESIDENT IN THE DISK DRIVE OR REQUIRE DOWNLINE LOADING FROM THE XXDP+ LOAD DEVICE. THESE DIAGNOSTIC PROGRAMS THAT RUN IN THE DISK DRIVES ARE NOT PART OF THIS DIAGNOSTIC PRODUCT, BUT ARE PRODUCED BY THE DISK DEVELOPMENT GROUP. THIS UDA DIAGNOSTIC PROGRAM ONLY KNOWS THE PROCEDURE TO EXECUTE THE DISK RESIDENT DIAGNOSTICS AND HOW TO DETERMINE WHETHER A TEST PASSED OR FAILED.

ONE AT A TIME, EACH UDA SELECTED FOR TESTING WILL BE INITIALIZED AND A 'DIAGNOSTIC MACHINE' PROGRAM DOWNLINE LOADED. THE 'DIAGNOSTIC MACHINE' PROGRAM WILL ASK WHAT DRIVES ARE TO BE TESTED, THEN WILL ISSUE SEVERAL ECHO FRAMES TO THE DISK DRIVE AND CHECK FOR THE CORRECT RESPONSE FROM THE DRIVE. THIS SHOULD SERVE AS A GOOD INDICATOR THAT THE UDA AND DISK DRIVE CAN COMMUNICATE.

A DIAGNOSE COMMAND WILL THEN BE ISSUED TO THE DRIVE TO REQUEST THE DRIVE RUN ALL OF ITS DIAGNOSTICS. IF THE DISK DRIVE REQUESTS A DOWNLINE LOAD OF A DRIVE DIAGNOSTIC, THE DIAGNOSTIC PROGRAM WILL BE READ FROM THE XXDP+ LOAD DEVICE, DOWNLINE LOAD THE FILE INTO THE DISK DRIVE AND START ITS EXECUTION. THERE IS NO LIMIT TO THE NUMBER OF DOWNLINE LOADS THAT CAN BE REQUESTED BY A DRIVE.

IF THE 'MANUAL INTERVENTION MODE' SOFTWARE QUESTION WAS ANSWERED NO (DEFAULT) TESTING WILL PROCEED TO THE NEXT DRIVE. WHEN ALL DRIVES ON THE UDA HAVE BEEN TESTED, THE NEXT UDA SELECTED FOR TESTING WILL THEN BE TESTED IN THE SAME MANNER. WHEN ALL UDA'S HAVE BEEN TESTED, TEST #2 WILL END.

IF THE 'MANUAL INTERVENTION MODE' SOFTWARE QUESTION WAS ANSWERED YES, AN INTERACTIVE MODE WILL BE ENTERED TO ALLOW THE OPERATOR TO PERFORM DIAGNOSTIC ACTIVITIES ON THE DISK DRIVE AS DESIRED. THE SERVICE MANUAL FOR THE DISK DRIVE MUST BE USED TO DETERMINE WHAT DIAGNOSTIC CAPABILITIES ARE AVAILABLE.

FIRST, A BRIEF DESCRIPTION OF AVAILABLE COMMANDS WILL BE PRINTED AS FOLLOWS:

TEST #2 MANUAL INTERVENTION ON UNIT XX UDA AT XXXXXX DRIVE XXX
TO WRITE AND READ MEMORY:
 W DATA REGION OFFSET
 R REGION OFFSET
TO RUN A DIAGNOSTIC:
 D REGION
TO EXIT QUESTIONING:

E
DATA, REGION AND OFFSET ARE HEX VALUES.
?

COMMANDS MAY BE TYPED AFTER THE QUESTION MARK PROMPT. EACH COMMAND WILL BE PROCESSED AS ENTERED AND RESULTS DISPLAYED IMMEDIATELY. THE EXIT COMMAND WILL ALLOW THE DIAGNOSTIC TO PROCEED.

READ AND WRITE COMMANDS WILL REMEMBER THE REGION AND OFFSET VALUES. SUCCESSIVE READ AND SUCCESSIVE WRITE COMMANDS WILL AUTOMATICALLY INCREMENT TO THE NEXT OFFSET IF THE REGION AND OFFSET VALUES ARE NOT TYPED. IF A REGION IS TYPED BUT NOT AN OFFSET, OFFSET ZERO WILL BE USED.

ONE TO FOUR BYTES OF DATA MAY BE ENTERED BY A SINGLE WRITE COMMAND, DEPENDING ON THE NUMBER OF DIGITS TYPED IN THE HEX VALUE. A READ COMMAND WILL ALWAYS RETURN FOUR BYTES OF DATA. EXAMPLES:

1. W FF FFFC 4
2. W 010203
3. R FFFC 0004
FFFC 0004/ FF 01 02 03
4. W F0F1F2F3 FFFC
5. R
FFFC 0000/ F0 F1 F2 F3
6. R
FFFC 0004/ FF 01 02 03

COMMAND 1 WRITES ONE BYTE (FF) INTO REGION FFFC, OFFSET 4. COMMAND 2 WRITES THREE BYTES (01, 02 AND 03) INTO THE SAME REGION WITH OFFSETS 5, 6 AND 7. COMMAND 3 READS FOUR BYTES STARTING AT REGION FFFC OFFSET 4. COMMAND 4 WRITES FOUR BYTES AT REGION FFFC OFFSET 0. COMMANDS 5 AND 6 READ THE EIGHT BYTES.

THE DIAGNOSE COMMAND WILL REMEMBER THE REGION FROM PREVIOUS DIAGNOSE COMMANDS ONLY, BECAUSE THE REGION CONTAINING THE DIAGNOSTIC IS GENERALLY NOT THE SAME REGION USED TO WRITE PARAMETERS OR READ RESULTS. IF THE DIAGNOSTIC RETURNS ANY DATA, THE DATA WILL BE PRINTED IMMEDIATELY.

TEST #3 - DISK FUNCTION TEST

THE PURPOSE OF TEST #3 IS TO FUNCTIONALLY TEST THE DISK DRIVE. ON A DRIVE THAT IS WELL DIAGNOSED BY ITS DISK RESIDENT DIAGNOSTICS (EXECUTED BY TEST #2) THESE FUNCTIONAL TESTS WILL HAVE LITTLE VALUE. ON A DRIVE THAT HAS NO OR MINIMAL RESIDENT DIAGNOSTICS, THESE FUNCTIONAL TESTS WILL HAVE MORE VALUE.

TEST #3 WILL START BY INITIALIZING EACH UDA SELECTED FOR TESTING AND THEN DOWNLINE LOAD A 'DIAGNOSTIC MACHINE' PROGRAM INTO EACH UDA. ONCE ALL UDAS HAVE BEEN STARTED, THE PDP-11 PROGRAM WILL RESPOND TO REQUESTS FROM ALL UDAS. WHEN ALL THE UDAS HAVE INDICATED THE END OF TESTING, TEST #3 WILL END.

THE 'DIAGNOSTIC MACHINE' PROGRAM WILL PERFORM THE FOLLOWING FUNCTIONS

ON EACH DRIVE:

1. ISSUE A DRIVE CLEAR COMMAND.
2. ISSUE INITIATE RECALIBRATE COMMAND.
3. ISSUE A CHANGE MODE COMMAND TO ENABLE DIAGNOSTIC CYLINDER ACCESS AND SET THE DRIVE TO 512 BYTE SECTOR SIZE.
4. ISSUE INITIATE SEEK COMMAND TO LAST DIAGNOSTIC CYLINDER.
5. READ ALL FACTORY FORMATTED SECTOR HEADERS. IF NO HEADERS ON A TRACK CAN BE READ, REPORT THE ERROR, OTHERWISE CONTINUE.
6. STARTING WITH CYLINDER 0, GROUP 0 AND INCREMENTING THROUGH EVERY GROUP ON THE DISK, SEEK TO THE SELECTED GROUP, READ A HEADER ON TRACK 0 AND THEN SEEK TO THE FACTORY FORMATTED DIAGNOSTIC CYLINDER. READ FROM THE DIAGNOSTIC CYLINDER TO VERIFY DISK POSITIONED CORRECTLY.
7. ISSUE A CHANGE MODE COMMAND TO ENABLE FORMATTING OPERATIONS.
8. FORMAT ALL WRITABLE DBNS IN 512 BYTE FORMAT.
9. WRITE AND READ SEVERAL DATA PATTERNS TO EACH WRITABLE DBN. REPORT AN ERROR IF ALL DBNS ON ONE TRACK HAVE AN ERROR.

TEST #4 - DISK EXERCISER

THE PURPOSE OF TEST #4 IS TO EXERCISE THE DISK DRIVES IN A MANNER SIMILAR TO NORMAL USAGE UNDER STANDARD OPERATING SYSTEMS. EXECUTION OF THIS TEST SHOULD GIVE AN INDICATION OF THE PERFORMANCE OF THE DISK DRIVE. THIS TEST MAY BE RUN FOR LONG OR SHORT PERIODS OF TIME, DEPENDING ON HOW THE SOFTWARE QUESTIONS ARE ANSWERED.

THESE ARE TWO MODES OF OPERATION FOR TEST #4:

1. DEFAULT OPERATION ON THE ENTIRE AREA SELECTED (CUSTOMER OR DIAGNOSTIC) WITH ALL PARAMETERS SELECTED FOR RANDOM OPERATION AS SHOWN BY DEFAULT ANSWERS BELOW.
2. MANUAL INTERVENTION MODE WHERE A NUMBER OF QUESTIONS ARE ASKED AND OPERATION IS CONTROLLED BY THEIR ANSWERS.

WHICH MODE IS ENTIRELY DETERMINED BY THE ANSWER TO THE FIRST SOFTWARE QUESTION ASKING, "ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS?" THIS QUESTION WOULD NORMALLY HAVE BEEN ANSWERED NO (DEFAULT) AND TESTING WILL BEGIN IMMEDIATELY. IF ANSWERED YES, THE FOLLOWING SERIES OF QUESTIONS WILL BE ASKED FOR EACH UNIT SELECTED FOR TESTING:

THE FOLLOWING QUESTIONS REFER TO UNIT XX UDA AT XXXXXX DRIVE XXX

THIS MESSAGE WILL IDENTIFY TO WHICH DRIVE THE QUESTIONS ARE BEING ASKED. THE ENTIRE SERIES OF QUESTIONS WILL BE ASKED FOR EACH DRIVE, THERE IS NO SHORT WAY TO ANSWER LIKE IN THE

HARDWARE QUESTIONS.

NUMBER OF BAD BLOCKS (D) 0 ?

AN ANSWER IN THE RANGE OF 1 TO 16 WILL ALLOW THAT MANY BAD BLOCK NUMBERS TO BE ENTERED. THE PROGRAM WILL ALLOW WRITES AND READS TO THESE BLOCKS BUT NO ERROR MESSAGES WILL BE PRINTED FOR THESE BLOCKS. ERRORS ENCOUNTERED ON THESE BLOCKS WILL NOT APPEAR IN THE STATISTICS. ANSWER ZERO TO BYPASS ENTERING BAD BLOCKS.

BAD BLOCK (A) ?

THIS QUESTION WILL BE ASKED THE NUMBER OF TIMES REQUESTED BY THE PREVIOUS ANSWER. ANY DECIMAL NUMBER THAT CAN BE CONVERTED INTO A 28-BIT BINARY VALUE WILL BE ACCEPTED. NO OTHER ERROR CHECKING WILL BE MADE AT THIS TIME TO DETERMINE IF THE BLOCK NUMBER ACTUALLY EXISTS ON THE DISK.

READ ONLY (L) N ?

ANSWER YES TO DICTATE READ ONLY AND PREVENT TEST #4 FROM PERFORMING ANY WRITES TO THE DISK. NOTE THAT TEST #3 WILL STILL WRITE TO THE DIAGNOSTIC CYLINDERS.

WRITE ONLY (L) N ?

THIS QUESTION WILL ONLY BE ASKED IF THE PREVIOUS QUESTION WAS ANSWERED NO. ANSWER YES TO DICTATE WRITE ONLY.

CHECK ALL WRITES BY READING (L) N ?

ANSWER YES TO CAUSE ALL WRITES TO BE CHECKED BY READING THE DATA IMMEDIATELY AFTER THE WRITE OPERATION.

RANDOMLY CHECK WRITES BY READING (L) Y ?

THIS QUESTION WILL ONLY BE ASKED IF THE PREVIOUS QUESTION WAS ANSWERED NO. ANSWER YES FOR THE WRITE CHECK TO BE PERFORMED RANDOMLY. ANSWER NO IF WRITE CHECKS ARE NOT DESIRED.

DATA PATTERN - 0 FOR RANDOM SELECTION (D) 0 ?

THERE ARE 16 DATA PATTERNS AVAILABLE, SELECTED AS 1 TO 16. PATTERN NUMBER 0 WILL CAUSE PATTERNS 1 TO 15 TO BE RANDOMLY SELECTED FOR EACH WRITE. IF PATTERN NUMBER 16 IS SELECTED, THE FOLLOWING SET OF QUESTIONS WILL BE ASKED FOR A PATTERN TO BE INPUT.

ENABLE ECC DATA CORRECTION (L) Y ?

A YES ANSWER WILL ENABLE THE USE OF ECC TO CORRECT DATA ERRORS. NO ERROR MESSAGE WILL BE PRINTED WHEN THE ECC PROPERLY CORRECTS THE DATA ON THE FIRST READ OF ANY DISK BLOCK. READ RETRIES WILL END WHEN A RE-READ PRODUCES AN ECC CORRECTABLE ERROR. THE USE OF ECC CORRECTION WILL APPEAR ONLY IN THE STATISTICAL REPORT FOR THE DRIVE.

A NO ANSWER WILL PREVENT THE USE OF ECC. ALL ECC ERRORS WILL CAUSE AN ERROR MESSAGE TO BE PRINTED AND RETRIES TO BE ATTEMPTED.

COMPARE ALL DATA READ (L) N ?

ANSWER YES TO CAUSE A DATA COMPARE AFTER EVERY READ.

RANDOMLY COMPARE DATA READ (L) Y ?

THIS QUESTION WILL ONLY BE ASKED IF THE PREVIOUS QUESTION WAS ANSWERED NO. ANSWER YES FOR THE DATA COMPARE TO BE PERFORMED ON RANDOM RECORDS. ANSWER NO IF DATA COMPARES ARE NOT DESIRED.

ENABLE RETRIES (L) Y

A YES ANSWER WILL ENABLE RETRIES TO BE PERFORMED ON DISK ERRORS.

RANDOM SEEK MODE (L) Y ?

ANSWER YES TO CAUSE BLOCK NUMBERS TO BE CHOSEN RANDOMLY. ANSWER NO TO CAUSE BLOCK NUMBERS TO BE SELECTED SEQUENTIALLY UP AND DOWN THE DISK SURFACE.

DO YOU WISH TO:

- 0 - TEST ENTIRE AREA SELECTED
- 1 - SPECIFY BEGIN/END SETS TO TEST
- 2 - SPECIFY TRACKS AND CYLINDERS TO TEST
- 3 - SPECIFY GROUPS AND CYLINDERS TO TEST
- 4 - SPECIFY CYLINDERS TO TEST

(D) 0 ?

THIS QUESTION SPECIFIES THE OPTIONS AVAILABLE TO LIMIT TESTING TO A PORTION OF THE SELECTED AREA (CUSTOMER OR DIAGNOSTIC) OF THE DISK. A ZERO ANSWER IS THE DEFAULT WHICH SPECIFIES TO USE THE ENTIRE AREA FOR THE TEST. OTHER ANSWERS WILL CAUSE ADDITIONAL QUESTIONS TO BE ASKED.

NUMBER OF BEGIN/END SETS (D) 1 ?

BEGIN BLOCK (A) 0 ?

END BLOCK (A) 0 ?

THESE QUESTIONS ARE ASKED IF BEGIN/END SETS WERE SELECTED TO LIMIT THE TESTING AREA (ANSWER 1). ONE TO FOUR SETS MAY BE SPECIFIED. THE BEGIN BLOCK AND END BLOCK QUESTIONS ARE ASKED AS MANY TIMES AS NEEDED.

NUMBER OF TRACKS TO TEST (D) 1 ?

TRACK (D) 0 ?

NUMBER OF GROUPS TO TEST (D) 1 ?

GROUP (D) 0 ?

ONE OF THESE SETS OF QUESTIONS IS ASKED IF EITHER TRACKS AND

CYLINDERS OR GROUPS AND CYLINDERS WAS SPECIFIED TO LIMIT THE TESTING AREA (ANSWERS 2 OR 3). UP TO SEVEN TRACKS OR GROUPS MAY BE SPECIFIED ON WHICH TESTING WILL BE LIMITED.

DO YOU WISH TO LIMIT THE CYLINDERS TESTED (L) N ?

THIS QUESTION IS ASKED ONLY AFTER THE TRACKS OR GROUPS HAVE BEEN SPECIFIED ABOVE. IF TESTING IS TO BE FURTHER LIMITED TO A SET OF CYLINDERS, ANSWER YES AND THE FOLLOWING TWO QUESTIONS WILL BE ASKED:

STARTING CYLINDER (A) 0 ?
ENDING CYLINDER (A) 0 ?

THESE QUESTIONS ARE ASKED IF THE QUESTION IMMEDIATELY ABOVE WAS ANSWERED YES OR IF CYLINDERS WERE SELECTED TO LIMIT THE TESTING AREA (ANSWER 4). ONE SET OF CYLINDER NUMBERS MAY BE SPECIFIED TO LIMIT THE TESTING AREA.

AFTER THE ABOVE QUESTIONS HAVE BEEN ASKED FOR ALL DRIVES SELECTED FOR TESTING, THE FOLLOWING QUESTIONS WILL BE ASKED IF DATA PATTERN 16 WAS SELECTED FOR ANY DRIVE:

NUMBER OF WORDS IN DATA PATTERN 16 (D) 1 ?
DATA WORD (O) 0 ?

DATA PATTERN 16 CAN BE INPUT BY THESE QUESTIONS. A DATA PATTERN CONSISTS OF A BUFFER OF ONE TO 16 WORDS WHICH IS REPEATED THROUGHOUT THE DATA PORTION OF THE DISK BLOCK. ENTER THE CONTENTS OF THE DATA PATTERN BUFFER. THE DATA WORD QUESTION WILL BE REPEATED AS NEEDED.

TEST #4 WILL START BY INITIALIZING EACH UDA SELECTED FOR TESTING AND THEN DOWNLINING A "DIAGNOSTIC MACHINE" PROGRAM INTO EACH UDA. THE "DIAGNOSTIC MACHINE" PROGRAM WILL ASK WHAT DRIVES ARE TO BE TESTED AND THEN WILL ASK FOR THE PARAMETERS FOR EACH DRIVE (THE ANSWERS TO THE MANUAL INTERVENTION QUESTIONS OR THEIR DEFAULTS). ONCE ALL UDAS HAVE BEEN STARTED, THE PDP-11 PROGRAM WILL RESPOND TO REQUESTS FROM ALL UDAS.

THE DISK WILL THEN BE EXERCISED ACCORDING TO THE PARAMETERS. THE EXERCISE CONSISTS OF SELECTING A DISK SECTOR, SEEKING TO THE PROPER CYLINDER, THEN READING OR WRITING THE SECTOR. THE PARAMETERS WILL CONTROL HOW THE DISK SECTOR IS SELECTED, WHETHER THE SECTOR IS WRITTEN OR READ AND WHETHER A WRITE IS FOLLOWED BY A READ (WRITE CHECK).

THE "DIAGNOSTIC MACHINE" PROGRAM WILL PERIODICALLY SEND STATISTICS TO THE PDP-11 PROGRAM. THESE STATISTICS WILL INCLUDE COUNTS OF READS, WRITES, SEEKS AND ERRORS ON A PER DRIVE BASIS. THE PDP-11 PROGRAM WILL ACCUMULATE THE STATISTICS FROM ALL THE UDAS AND WATCH FOR THE TRANSFER LIMIT TO BE EXCEEDED. AS LONG AS THE ERROR LOG IS NOT ENABLED, THE EXCEEDING OF THE TRANSFER LIMIT WILL CAUSE THE END OF TEST #4.

EACH TIME AN ERROR OCCURS, THE "DIAGNOSTIC MACHINE" WILL TELL THE PDP-11 PROGRAM. A MESSAGE WILL BE PRINTED (OR STORED IN THE LOG BUFFER) AND THEN THE ERROR LIMIT FOR THE DRIVE WILL BE CHECKED. IF THE

ERROR LIMIT HAS BEEN REACHED, THE DRIVE WILL BE DROPPED FROM TESTING. IF NO MORE DRIVES REMAIN TO BE TESTED, TEST #4 WILL END (UNLESS THE ERROR LOG IS ENABLED).

WHEN THE END OF TEST #4 OCCURS, THE ACCUMULATED STATISTICS FOR EACH DRIVE WILL BE PRINTED. THIS STATISTICAL REPORT CAN BE PRINTED AT ANY TIME DURING TEST #4 BY TYPING CONTROL-C THEN THE PRINT COMMAND.

THE DATA PATTERNS TO BE USED BY TEST #4 ARE INDICATED BELOW. EACH PATTERN IS GENERATED BY WRITING THE PATTERN NUMBER IN EACH 4-BIT NIBBLE OF THE FIRST WORD, THEN REPEATING THE DATA PATTERN (SEQUENCE OF ONE TO 16 WORDS) THROUGHOUT THE REST OF THE DATA BUFFER. PATTERN NUMBER 16 WRITES NIBBLES OF ZEROS. WHEN PATTERN NUMBER ZERO IS USED, THE ACTUAL PATTERN NUMBER WRITTEN (1 TO 15) IS PLACED IN THE NIBBLES.

PATTERN 0 THIS PATTERN NUMBER IS USED TO INDICATE ANY PATTERN NUMBER 1 TO 15 CHOSEN AT RANDOM.

PATTERN 1 WORDS IN PATTERN SEQUENCE - 1

SEQUENCE (OCTAL) 105613
SEQUENCE (HEX) 8B8B

PATTERN 2 WORDS IN PATTERN SEQUENCE - 1

SEQUENCE (OCTAL) 031463
SEQUENCE (HEX) 3333

PATTERN 3 WORDS IN PATTERN SEQUENCE - 1

SEQUENCE (OCTAL) 030221
SEQUENCE (HEX) 3091

PATTERN 4 WORDS IN PATTERN SEQUENCE - 16 (SHIFTING ONES)

SEQUENCE (OCTAL) 000001, 000003, 000007, 000017, 000037,
000077, 000177, 000377, 000777, 001777,
003777, 007777, 017777, 037777, 077777,
177777

SEQUENCE (HEX) 0001, 0003, 0007, 000F, 001F, 003F,
007F, 00FF, 01FF, 03FF, 07FF, 0FFF,
1FFF, 3FFF, 7FFF, FFFF

PATTERN 5 WORDS IN PATTERN SEQUENCE - 16 (SHIFTING ZEROS)

SEQUENCE (OCTAL) 177776, 177774, 177770, 177760, 177740,
177700, 177600, 177400, 177000, 176000,
174000, 170000, 160000, 140000, 100000,
000000

SEQUENCE (HEX) FFFE, FFFC, FFF8, FFF0, FFE0, FFC0,
FF80, FF00, FE00, FC00, F800, F000,
E000, C000, 8000, 0000

PATTERN 6 WORDS IN PATTERN SEQUENCE - 16

SEQUENCE (OCTAL) 000000, 000000, 000000, 177777, 177777,
177777, 000000, 000000, 177777, 177777,
000000, 177777, 000000, 177777, 000000,
177777

SEQUENCE (HEX) 0000, 0000, 0000, FFFF, FFFF, FFFF,
0000, 0000, FFFF, FFFF, 0000, FFFF,
0000, FFFF, 0000, FFFF

PATTERN 7 WORDS IN PATTERN SEQUENCE - (BINARY 1011011011011001)

SEQUENCE (OCTAL) 133331
SEQUENCE (HEX) B6D9

PATTERN 8 WORDS IN PATTERN SEQUENCE - 16

SEQUENCE (OCTAL) 052525, 052525, 052525, 125252, 125252,
125252, 052525, 052525, 125252, 125252,
052525, 125252, 052525, 125252, 052525,
125252

SEQUENCE (HEX) 5555, 5555, 5555, AAAA, AAAA, AAAA,
5555, 5555, AAAA, AAAA, 5555, AAAA,
5555, AAAA, 5555, AAAA

PATTERN 9 WORDS IN PATTERN SEQUENCE - 1 (BINARY 1101101101101100)

SEQUENCE (OCTAL) 155554
SEQUENCE (HEX) D86C

PATTERN 10 WORDS IN PATTERN SEQUENCE - 16

SEQUENCE (OCTAL) 026455, 026455, 026455, 151322, 151322,
151322, 026455, 026455, 151322, 151322,
026455, 151322, 026455, 151322, 026455,
151322

SEQUENCE (HEX) 2D2D, 2D2D, 2D2D, D2D2, D2D2, D2D2,
2D2D, 2D2D, D2D2, D2D2, 2D2D, D2D2,
2D2D, D2D2, 2D2D, D2D2

PATTERN 11 WORDS IN PATTERN SEQUENCE - 1 (BINARY 0110110110110110)

SEQUENCE (OCTAL) 066666
SEQUENCE (HEX) 6DD6

PATTERN 12 WORDS IN PATTERN SEQUENCE - 16 (RIPPLE ONE)

SEQUENCE (OCTAL) 000001, 000002, 000004, 000010, 000020,
000040, 000100, 000200, 000400, 001000,
002000, 004000, 010000, 020000, 040000,
100000

SEQUENCE (HEX) 0001, 0002, 0004, 0008, 0010, 0020,
0040, 0080, 0100, 0200, 0400, 0800,
1000, 2000, 4000, 8000

PATTERN 13 WORDS IN PATTERN SEQUENCE - 16 (RIPPLE ZERO)
SEQUENCE (OCTAL) 177776, 177775, 177773, 177767, 177757,
177737, 177677, 177577, 177377, 176777,
175777, 173777, 167777, 157777, 137777,
077777

SEQUENCE (HEX) FFFE, FFFD, FFFB, FFF7, FFEF, FFDF,
FFBF, FF7F, FEFF, FDFF, FBFF, F7FF,
EFFF, DFFF, BFFF, 7FFF

PATTERN 14 WORDS IN PATTERN SEQUENCE - 3

SEQUENCE (OCTAL) 155555, 133333, 155555
SEQUENCE (HEX) DB6D, B6DB, DB6D

PATTERN 15 WORDS IN PATTERN SEQUENCE - 16

SEQUENCE (OCTAL) 133331, 133331, 133331, 155554, 155554,
155554, 133331, 133331, 155554, 155554,
133331, 155554, 133331, 155554, 133331,
155554

SEQUENCE (HEX) B6D9, B6D9, B6D9, DB6C, DB6C, DB6C,
B6D9, B6D9, DB6C, DB6C, B6D9, DB6C,
B6D9, DB6C, B6D9, DB6C

PATTERN 16 THIS IS THE OPERATOR SELECTABLE PATTERN IN MANUAL
INTERVENTION MODE. QUESTIONS ARE ASKED WHEN TEST #4 IS
STARTED FOR THE OPERATOR TO INPUT THE NUMBER OF WORDS IN
THE SEQUENCE AND THE CONTENTS OF THE WORDS.

9
 10
 36
 38 000000
 39
 40 002000
 42
 43 002000
 44
 45
 46
 47 000000
 48 000000
 49
 50
 51
 52
 53
 54
 55 002000
 56
 73
 77 002000
 002000
 002000 103
 002001 132
 002002 104
 002003 125
 002004 101
 002005 000
 002006 000
 002007 000
 002010
 002010 101
 002011
 002011 060
 002012
 002012 000001
 002014
 002014 000000
 002016
 002016 033572
 002020
 002020 034036
 002022
 002022 002136
 002024
 002024 002154
 002026
 002026 034622
 002030
 002030 000000
 002032
 002032 000000
 002034
 002034 000001
 002036

.TITLE CZUDCAO UDA AND DRIVE DIAGNOSTIC
 .SBTTL PROGRAM HEADER

.ENABL ABS
 .ENABL AMA
 = 2000

BGNMOD

;ASSEMBLY CONTROL

ENG=0 ;SET NON-ZERO TO ASSEMBLE ENGINEERING CODE
 MFG=0 ;SET NON-ZERO TO ASSEMBLE MANUFACTURING CODE

;++
 ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
 ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
 :--

POINTER BGNRPT, BGNSW, BGNSFT, ERR TBL, BGNSETUP

HEADER CZDUA,A,0,0,1,PRI07

L\$NAME::
 .ASCII /C/
 .ASCII /Z/
 .ASCII /D/
 .ASCII /U/
 .ASCII /A/
 .BYTE 0
 .BYTE 0
 .BYTE 0
 L\$REV::
 .ASCII /A/
 L\$DEPO::
 .ASCII /0/
 L\$UNIT::
 .WORD T\$PTHV
 L\$TIML::
 .WORD 0
 L\$HPCP::
 .WORD L\$HARD
 L\$SPCP::
 .WORD L\$SOFT
 L\$HPTP::
 .WORD L\$HW
 L\$SPTP::
 .WORD L\$SW
 L\$LADP::
 .WORD L\$LAST
 L\$STA::
 .WORD 0
 L\$CO::
 .WORD 0
 L\$DTYP::
 .WORD 1
 L\$APT::

PROGRAM HEADER

002036 000000
 002040
 002040 002124
 002042
 002042 000340
 002044
 002044 000000
 002046
 002046 000000
 002050
 002050 003
 002051 003
 002052
 002052 000000
 002054 000000
 002056
 002056 000000
 002060
 002060 002434
 002062
 002062 024602
 002064
 002064 000000
 002066
 002066 000000
 002070
 002070 000000
 002072
 002072 000000
 002074
 002074 000000
 002076
 002076 002460
 002100
 002100 104035
 002102
 002102 002162
 002104
 002104 026024
 002106
 002106 027476
 002110
 002110 027474
 002112
 002112 026016
 002114
 002114 000000
 002116
 002116 000000
 002120
 002120 000000

LSDTP:: .WORD 0
 L\$PRIO:: .WORD L\$DISPATCH
 L\$ENVI:: .WORD PRI07
 L\$EXP1:: .WORD 0
 L\$MREV:: .WORD 0
 L\$EF:: .BYTE C\$REVISION
 .BYTE C\$EDIT
 L\$SPC:: .WORD 0
 L\$DEVP:: .WORD 0
 L\$REPP:: .WORD L\$DVTYP
 L\$EXP4:: .WORD L\$RPT
 L\$EXP5:: .WORD 0
 L\$AUT:: .WORD 0
 L\$DUT:: .WORD 0
 L\$LUN:: .WORD 0
 L\$DESP:: .WORD 0
 L\$LOAD:: .WORD L\$DESC
 EMT E\$LOAD
 L\$ETP:: .WORD L\$ERRTBL
 L\$IICP:: .WORD L\$INIT
 L\$CCP:: .WORD L\$CLEAN
 L\$ACP:: .WORD L\$AUTO
 L\$PRT:: .WORD L\$PROT
 L\$TEST:: .WORD 0
 L\$DLY:: .WORD 0
 L\$HIME:: .WORD 0

DISPATCH TABLE

1
2
3
4
5
6
7
8
9

.SBTTL DISPATCH TABLE

;++
: THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
: IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
:--

DISPATCH 4

002122
002122 000004
002124
002124 027504
002126 030710
002130 031006
002132 031044

.WORD 4
L\$DISPATCH::
.WORD T1
.WORD T2
.WORD T3
.WORD T4

1
2
3
4
5
6
7
8
9
10
11
21
22
23
24
25
26
27

.SBTTL DEFAULT HARDWARE P-TABLE

;++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
: AND IS USED AS A 'TEMPLATE' FOR BUILDING THE P-TABLES.
:--

002134
002134 000006
002136
002136

BGNHW DFPTBL

.WORD L10000-L\$HW/2
L\$HW::
DFPTBL::

002136 172150
002140 000154
002142 000005
002144 000000
002146 000000
002150 000000
002152
002152

.WORD 172150
.WORD 154
.WORD 5.
.WORD 0.
.WORD 0.
.WORD 0.
ENDHW

: UNIBUS ADDRESS
: VECTOR ADDRESS
: BR LEVEL
: UNIBUS BURST RATE
: LOGICAL DRIVE NUMBER
: CUSTOMER DATA AREA

L10000:

```
1          .SBTTL  SOFTWARE P-TABLE
2
3
4          :++
5          : THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
6          : PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
7          : SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
8          : AT RUN TIME.
9          :--
10         BGNSW  SFPTBL
11
12         002152      000003
13         002152
14         002154
15         002154
16
17
18
19
20
21         002154      000040
22         002156      000000
23         002160      040400
24
25         002162
26         002162
27
28
29         002162
30
31         ENDSW
32
33         ENDMOD
34
35         .SBTTL
```

.WORD L10001-L\$\$W/2
L\$\$W::
SFPTBL::

:OFFSET USE
: 0. ERROR LIMIT
: 2. DATA TRANSFER LIMIT (MEGABITS)
: 4. SINGLE BIT QUESTIONS

L10001:

8
9
37
47
48 002162
49
50
51
52
53
54
69
70 002162

:.TITLE GLOBAL AREAS (3)
:SBTTL GLOBAL EQUATES SECTION

BGNMOD

:.++
: THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
: ARE USED IN MORE THAN ONE TEST.
:--

EQUALS

: BIT DIFINITIONS

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

001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

: EVENT FLAG DEFINITIONS
: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	: START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	: RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	: CONTINUE COMMAND WAS ISSUED
000035	EF.NEW== 29.	: A NEW PASS HAS BEEN STARTED
000034	EF.PWR== 28.	: A POWER-FAIL/POWER-UP OCCURRED

: PRIORITY LEVEL DEFINITIONS

000340 PRI07== 340

```
000300 PRI06== 300
000240 PRI05== 240
000200 PRI04== 200
000140 PRI03== 140
000100 PRI02== 100
000040 PRI01== 40
000000 PRI00== 0
:
: OPERATOR FLAG BITS
:
000004 EVL== 4
000010 LOT== 10
000020 ADR== 20
000040 IDU== 40
000100 ISR== 100
000200 UAM== 200
000400 BOE== 400
001000 PNT== 1000
002000 PRI== 2000
004000 IXE== 4000
010000 IBE== 10000
020000 IER== 20000
040000 LOE== 40000
100000 HOE== 100000
```

```

1          ;UDA BIT DEFINITIONS
2
3          ;UDASA REGISTER UNIVERSAL READ BITS
4
5          004000 SA.S1= 004000          ;STEP 1 STATUS BIT
6          010000 SA.S2= 010000          ;STEP 2 STATUS BIT
7          020000 SA.S3= 020000          ;STEP 3 STATUS BIT
8          040000 SA.S4= 040000          ;STEP 4 STATUS BIT
9          100000 SA.ERR= 100000         ;ERROR INDICATOR
10
11         ;UDASA REGISTER ERROR STATUS BITS
12
13         003777 SA.ERC= 003777         ;ERROR CODE
14
15         ;UDASA REGISTER STEP ONE READ BITS
16
17         002000 SA.NV= 002000         ;NON SETTABLE INTERRUPT VECTOR
18         001000 SA.A2= 001000         ;22 BIT ADDRESS BUS
19         000400 SA.DI= 000400         ;ENHANCED DIAGNOSTICS
20         ;      000377                 ;ALL BITS RESERVED
21
22         ;UDASA REGISTER STEP ONE WRITE BITS
23
24         000177 SA.VEC= 000177         ;INTERRUPT VECTOR (DIVIDED BY 4)
25         000200 SA.INT= 000200         ;INTERRUPT ENABLE DURING INITIALIZATION
26         003400 SA.MSG= 003400         ;MESSAGE RING LENGTH
27         034000 SA.CMD= 034000         ;COMMAND RING LENGTH
28         ;      040000                 ;RESERVED
29         100000 SA.STP= 100000        ;STEP - MUST ALWAYS BE WRITTEN A ONE
30
31         ;UDASA REGISTER STEP TWO READ BITS
32
33         000007 SA.MSE= 000007         ;MESSAGE RING LENGTH ECHO
34         000070 SA.CME= 000070         ;COMMAND RING LENGTH ECHO
35         ;      000100                 ;RESERVED
36         000200 SA.STE= 000200         ;STEP ECHO
37         003400 SA.CTP= 003400         ;CONTROLLER TYPE
38
39         ;UDASA REGISTER STEP TWO WRITE BITS
40
41         000001 SA.PRG= 000001         ;ENABLE VAX UNIBUS ADAPTER PURGE INTERRUPT
42         ;      177776                 ;LOW ORDER MESSAGE RING BYTE ADDRESS
43
44         ;UDASA REGISTER STEP THREE READ BITS
45
46         000177 SA.VCE= 000177         ;INTERRUPT VECTOR ECHO
47         000200 SA.INE= 000200         ;INTERRUPT ENABLE ECHO
48         000400 SA.NVE= 000400         ;VECTOR NOT PROGRAMMABLE
49         ;      003000                 ;RESERVED
50
51         ;UDASA REGISTER STEP THREE WRITE BITS
52
53         ;      077777                 ;HIGH ORDER MESSAGE RING BYTE ADDRESS
54         100000 SA.TST= 100000        ;PURGE POLE TEST ENABLE
55
56         ;UDASA REGISTER STEP FOUR READ BITS
57
    
```


58	000377	SA.MCV= 000377	;UDA MICROCODE VERSION
59		; 003400	;RESERVED
60			
61		;UDASA REGISTER STEP FOUR WRITE BITS	
62			
63	000001	SA.GO= 000001	;GO BIT TO START UDA FIRMWARE
64	000002	SA.LFC= 000002	;LAST FAILURE CODE REQUEST
65	000374	SA.BST= 000374	;BURST LEVEL

```

1          ;COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS
2
3          100000          RG.OWN= 100000          ;SET WHEN UDA OWNS RING
4          040000          RG.FLG= 040000          ;FLAG BIT
5
6          ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
7          ;AND TWO PACKET AND BUFFER AREAS.
8
9          000004          HC.ISZ= 4.              ;SIZE OF INTERRUPT INDICATOR WORDS
10         000004          HC.RSZ= 4.              ;SIZE OF RING IN BYTES
11         000004          HC.ESZ= 4.              ;SIZE OF ENVELOPE WORDS BEFORE PACKET
12         000060          HC.PSZ= 48.             ;SIZE OF COMMAND AND MESSAGE PACKETS
13         000074          HC.BSZ= 60.             ;SIZE OF BUFFER
14
15         000000          HC.INT= 0.              ;INTERRUPT INDICATOR WORDS START
16         000004          HC.MSG= HC.INT+HC.ISZ    ;MESSAGE RING START
17         000006          HC.MCT= HC.MSG+2.        ;MESSAGE RING CONTROL WORD
18         000010          HC.CMD= HC.MSG+HC.RSZ    ;COMMAND RING START
19         000012          HC.CCT= HC.CMD+2.        ;COMMAND RING CONTROL WORDS
20         000014          HC.MEV= HC.CMD+HC.RSZ    ;MESSAGE ENVELOPE START
21         000020          HC.MPK= HC.MEV+HC.ESZ    ;MESSAGE PACKET START
22         000100          HC.CEV= HC.MPK+HC.PSZ    ;COMMAND ENVELOPE START
23         000104          HC.CPK= HC.CEV+HC.ESZ    ;COMMAND PACKET START
24         000164          HC.BF1= HC.CPK+HC.PSZ    ;FIRST BUFFER
25         000260          HC.BF2= HC.BF1+HC.BSZ    ;SECOND BUFFER
26
27         000354          HC.SIZ= HC.BF2+HC.BSZ    ;TOTAL SIZE OF HOST COMM AREA
28
29         ;VIRTUAL CIRCUIT IDENTIFIERS
30
31         000000          MSCP= 0                  ;MSCP CIRCUIT
32         000001          LOG= 1                   ;LOG CIRCUIT
33         177777          DIAG= -1                 ;DIAGNOSTIC CIRCUIT
34         001000          DUP= 1000                ;DIAGNOSTIC AND UTILITIES PROTOCOL
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

HC.INT	INTERRUPT INDICATORS	4 BYTES
HC.MSG HC.MCT	MESSAGE RING	4 BYTES
HC.CMD HC.CCT	COMMAND RING	4 BYTES
HC.MEV HC.MPK	MESSAGE ENVELOPE	52 BYTES
HC.CEV HC.CPK	COMMAND ENVELOPE	52 BYTES
HC.BF1	BUFFER # 1 (RESPONSE TO DM PROGRAM)	60 BYTES
HC.BF2	BUFFER # 2 (REQUEST FROM DM PROGRAM)	60 BYTES

```

1          ;COMMAND PACKET OPCODES
2
3          000001      OP.ABO= 1          :ABORT COMMAND
4          000020      OP.ACC= 20         :ACCESS COMMAND
5          000010      OP.AVL= 10         :AVAILABLE COMMAND
6          000021      OP.CCD= 21         :COMPARE CONTROLLER DATA COMMAND
7          000040      OP.CMP= 40         :COMPARE HOST DATA COMMAND
8          000022      OP.ERS= 22         :ERASE COMMAND
9          000023      OP.FLU= 23         :FLUSH COMMAND
10         000002      OP.GCS= 2          :GET COMMAND STATUS COMMAND
11         000003      OP.GUS= 3          :GET UNIT STATUS COMMAND
12         000011      OP.ONL= 11         :ONLINE COMMAND
13         000041      OP.RD= 41          :READ COMMAND
14         000024      OP.RPL= 24         :REPLACE COMMAND
15         000004      OP.SCC= 4          :SET CONTROLLER CHARACTERISTICS COMMAND
16         000012      OP.SUC= 12         :SET UNIT CHARACTERISTICS COMMAND
17         000042      OP.WR= 42          :WRITE COMMAND
18         000030      OP.MRD= 30         :MAINTENANCE READ COMMAND
19         000031      OP.MWR= 31         :MAINTENANCE WRITE COMMAND
20         000200      OP.END= 200        :END PACKET FLAG
21         000007      OP.SEX= 7          :SERIOUS EXCEPTION END PACKET
22         000100      OP.AVA= 100        :AVAILABLE ATTENTION MESSAGE
23         000101      OP.DUP= 101        :DUPLICATE UNIT NUMBER ATTENTION MESSAGE
24         000102      OP.SHC= 102        :SHADOW COPY COMPLETE ATTENTION MESSAGE
25         000103      OP.RLC= 103        :RESET COMMAND LIMIT ATTENTION MESSAGE
26
27         000001      OP.GSS= 1          :DUP GET STUD STATUS
28         000002      OP.ESP= 2          :DUP EXECUTE SUPPLIED PROGRAM
29         000003      OP.ELP= 3          :DUP EXECUTE LOCAL PROGRAM
30         000004      OP.SSD= 4          :DUP SEND STUD DATA
31         000005      OP.RSD= 5          :DUP RECEIVE STUD DATA
32
33         ;NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
34         ;PACKET FLAG TO THE COMMAND OPCODE. FOR EXAMPLE, A READ COMMAND'S END PACKET
35         ;CONTAINS THE VALUE OP.RD+OP.END IN ITS OPCODE FIELD. THE INVALID COMMAND END
36         ;PACKET CONTAINS JUST THE END PACKET FLAG (I.E., OP.END) IN ITS OPCODE FIELD.
37         ;THE SERIOUS EXCEPTION END PACKET CONTAINS THE SUM OF THE END PACKET FLAG
38         ;PLUS THE SERIOUS EXCEPTION OPCODE SHOWN ABOVE (I.E., OP.SEX+OP.END) IN ITS
39         ;OPCODE FIELD.
40         ;
41         ;COMMAND OPCODE BITS 3 THROUGH 5 INDICATE THE COMMAND CLASS, WHICH IS ENCODED
42         ;AS FOLLOWS:
43         ; 000 IMMEDIATE COMMANDS
44         ; 001 SEQUENTIAL COMMANDS
45         ; 010 NON-SEQUENTIAL COMMANDS THAT DO NOT INCLUDE A BUFFER DESCRIPTOR
46         ; 100 NON-SEQUENTIAL COMMANDS THAT DO INCLUDE A BUFFER DESCRIPTOR
    
```

```

1          ;COMMAND MODIFIERS
2
3          ;      = 020000
4          040000 MD.CMP= 040000      ;CLEAR SERIOUS EXCEPTION
5          100000 MD.EXP= 100000      ;COMPARE
6          010000 MD.ERR= 010000      ;EXPRESS REQUEST
7          004000 MD.SCH= 004000      ;FORCE ERROR
8          002000 MD.SCL= 002000      ;SUPPRESS CACHING (HIGH SPEED)
9          000100 MD.SEC= 000100      ;SUPPRESS CACHING (LOW SPEED)
10         000400 MD.SER= 000400      ;SUPPRESS ERROR CORRECTION
11         000200 MD.SSH= 000200      ;SUPPRESS ERROR RECOVERY
12         000100 MD.WBN= 000100      ;SUPPRESS SHADOWING
13         000400 MD.WBV= 000400      ;WRITE-BACK (NON-VOLATILE)
14         000020 MD.SEQ= 000020      ;WRITE BACK (VOLATILE)
15         000001 MD.SPD= 000001      ;WRITE SHADOW SET ONE UNIT AT A TIME
16         000001 MD.FEU= 000001      ;SPIN-DOWN
17         000002 MD.VOL= 000002      ;FLUSH ENTIRE UNIT
18         000001 MD.NXU= 000001      ;VOLATILE ONLY
19         000001 MD.RIP= 000001      ;NEXT UNIT
20         000002 MD.IMF= 000002      ;ALLOW SELF DESTRUCTION
21         000004 MD.SWP= 000004      ;IGNORE MEDIA FORMAT ERROR
22         000010 MD.CWB= 000010      ;SET WRITE PROTECT
23         000001 MD.PRI= 000001      ;CLEAR WRITE-BACK DATA LOST
24                                     ;PRIMARY REPLACEMENT BLOCK
25         ;END PACKET FLAGS
26
27         000200 EF.BBR= 000200      ;BAD BLOCK REPORTED
28         000100 EF.BBU= 000100      ;BAD BLOCK UNREPORTED
29         000040 EF.LOG= 000040      ;ERROR LOG GENERATED
30         000020 EF.SEX= 000020      ;SERIOUS EXCEPTION
31
32         ;CONTROLLER FLAGS
33
34         000200 CF.ATN= 000200      ;ENABLE ATTENTION MESSAGES
35         000100 CF.MSC= 000100      ;ENABLE MISCELLANEOUS ERROR LOG MESSAGES
36         000040 CF.OTH= 000040      ;ENABLE OTHER HOST'S ERROR LOG MESSAGES
37         000020 CF.THS= 000020      ;ENABLE THIS HOST'S ERROR LOG MESSAGES
38         000002 CF.SHD= 000002      ;SHADOWING
39         000001 CF.576= 000001      ;576 BYTE SECTORS
    
```



```

1          ;END PACKET OFFSETS
2
3          ;
4          000000      P.CRF= 0.          ;COMMAND REFERENCE NUMBER
5          000004      P.UNIT= 4.         ;UNIT NUMBER
6          000010      P.OPCD= 8.         ;OPCODE (ALSO CALLED ENDCODE)
7          000011      P.FLGS= 9.         ;END PACKET FLAGS
8          000012      P.STS= 10.        ;STATUS
9          000014      P.BCNT= 12.        ;BYTE COUNT
10         000034      P.FBBK= 28.        ;FIRST BAD BLOCK
11
12         ;
13         000014      P.OTRF= 12.        ;GET COMMAND STATUS END PACKET OFFSETS:
14         000020      P.CMST= 16.        ;OUTSTANDING REFERENCE NUMBER
15                                     ;COMMAND STATUS
16         ;
17         000014      P.MLUN= 12.        ;GET UNIT STATUS END PACKET OFFSETS:
18         000016      P.UNFL= 14.        ;MULTI-UNIT CODE
19         000020      P.HSTI= 16.        ;UNIT FLAGS
20         000024      P.UNTI= 20.        ;HOST IDENTIFIER
21         000034      P.MEDI= 28.        ;UNIT IDENTIFIER
22         000040      P.SHUN= 32.        ;MEDIA TYPE IDENTIFIER
23         000042      P.SHST= 34.        ;SHADOW UNIT
24         000044      P.TRCK= 36.        ;SHADOW STATUS
25         000046      P.GRP= 38.         ;TRACK SIZE
26         000050      P.CYL= 40.         ;GROUP SIZE
27         000054      P.RCTS= 44.        ;CYLINDER SIZE
28         000056      P.RBNS= 46.        ;RCT TABLE SIZE
29         000057      P.RCTC= 47.        ;RBNS / TRACK
30                                     ;RCT COPIES
31         ;
32         ;
33         000014      P.MLUN= 12.        ;ONLINE AND SET UNIT CHARACTERISTICS END PACKET AND AVAILABLE
34         000016      P.UNFL= 14.        ;ATTENTION MESSAGE OFFSETS:
35         000020      P.HSTI= 16.        ;MULTI-UNIT CODE
36         000024      P.UNTI= 20.        ;UNIT FLAGS
37         000034      P.MEDI= 28.        ;HOST IDENTIFIER
38         000040      P.SHUN= 32.        ;UNIT IDENTIFIER
39         000042      P.SHST= 34.        ;MEDIA TYPE IDENTIFIER
40         000044      P.UNCL= 36.        ;SHADOW UNIT
41         000050      P.UNSZ= 40.        ;SHADOW STATUS
42         000054      P.VSER= 44.        ;UNIT COMMAND LIMIT
43                                     ;UNIT SIZE
44                                     ;VOLUME SERIAL NUMBER
45         ;
46         000014      P.VRSN= 12.        ;SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS:
47         000016      P.CNTF= 14.        ;MSCP VERSION
48         000020      P.CTMO= 16.        ;CONTROLLER FLAGS
49         000022      P.CNCL= 18.        ;CONTROLLER TIMEOUT
          000024      P.CNTI= 20.        ;CONTROLLER COMMAND LIMIT
          ;CONTROLLER ID
    
```

```
1          ;STATUS AND EVENT CODE DEFINITIONS
2
3          000037      ST.MSK= 37          ;STATUS / EVENT CODE MASK
4          000040      ST.SUB= 40         ;SUB-CODE MULTIPLIER
5          000000      ST.SUC= 0          ;SUCCESS
6          000001      ST.CMD= 1          ;INVALID COMMAND
7          000002      ST.ABO= 2          ;COMMAND ABORTED
8          000003      ST.OFL= 3          ;UNIT-OFFLINE
9          000004      ST.AVL= 4          ;UNIT-AVAILABLE
10         000005      ST.MFE= 5          ;MEDIA FORMAT ERROR
11         000006      ST.WPR= 6          ;WRITE PROTECTED
12         000007      ST.CMP= 7          ;COMPARE ERROR
13         000010      ST.DAT= 10         ;DATA ERROR
14         000011      ST.HST= 11         ;HOST BUFFER ACCESS ERROR
15         000012      ST.CNT= 12         ;CONTROLLER ERROR
16         000013      ST.DRV= 13         ;DRIVE ERROR
17         000037      ST.DIA= 37         ;MESSAGE FROM AN INTERNAL DIAGNOSTIC
```



```

1          ;CONTROLLER TABLE DEFINITIONS
2          ;
3          ;ONE TABLE WILL BE SET UP BY INITIALIZE SECTION FOR EACH UDA SELECTED
4          ;FOR TESTING. TABLES ARE CONTIGUOUS. THE END OF THE TABLES IS
5          ;MARKED BY A WORD OF ZEROS.
6          ;
7          ;THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF CTABS.
8          ;THE NUMBER OF TABLES IS CONTAINED IN CTRLRS.
9
10         000000 C.UADR= 0.          ;UNIBUS ADDRESS OF UDAIP REGISTER
11         000002 C.UNIT= 2.         ; LOGICAL UNIT NUMBER (FIRST)
12         000077          CT.UNT= 000077 ; SET WHEN NOT AVAILABLE FOR TESTING
13         100000          CT.AVL= BIT15
14         000004 C.VEC= 4.          ; VECTOR ADDRESS
15         000777          CT.VEC= 000777 ; BR LEVEL
16         007000          CT.BRL= 007000 ; BURST LEVEL
17         000006 C.BST= 6.          ; INTERRUPT SERVICE ROUTINE FOR CONTROLLER
18         000010 C.JSR= 8.          ; THESE TWO WORDS LOADED WITH [JSR R0,UDASRV]
19         000012 C.JAD= 10.         ;
20         000014 C.FLG= 12.         ; FLAGS
21         000002          CT.RN= BIT1  ; DM PROGRAM RUNNING
22         000004          CT.CMD= BIT2 ; COMMAND ISSUED, WAITING FOR RESPONSE
23         000010          CT.MSG= BIT3 ; MESSAGE RESPONSE RECEIVED
24
25         000020          CT.REQ= BIT4 ; WHENEVER THIS BIT IS SET, CT.CMD IS CLEARED
26
27
28         000016 C.RING= 14.         ; BUFFER HAS BEEN GIVEN TO UDA FOR REQUEST
29         000020 C.DR0= 16.         ; SET WHENEVER READ STUD DATA COMMAND
30         000022 C.DR1= 18.         ; GIVEN TO UDA
31         000024 C.DR2= 20.         ; RING BUFFER ADDRESS
32         000026 C.DR3= 22.         ; POINTER TO DRIVE TABLES
33         000030 C.DR4= 24.         ; IF ZERO, NO DRIVE TABLE EXISTS
34         000032 C.DR5= 26.
35         000034 C.DR6= 28.
36         000036 C.DR7= 30.
37         000040 C.TO= 32.          ; TIMEOUT COUNTER
38         000042 C.TOH= 34.         ; (TWO WORDS)
39         000044 C.REF= 36.         ; COMMAND REFERENCE NUMBER
40
41         000046 C.SIZE= 38.        ; SIZE OF CONTROLLER TABLE IN BYTES
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

:DRIVE TABLE DEFINITIONS

:ONE DRIVE TABLE WILL BE SET UP BY THE INITIALIZE SECTION FOR EACH
 :DRIVE SELECTED FOR TESTING. EACH TABLE IS POINTED TO BY A
 :WORD IN THE CONTROLLER TABLE ON WHICH THE DRIVE EXISTS.

000000
000002
000077
100000
000004
040000
020000
010000
004000
002000
001000
000400
000100
000040
000020
000010
000004
000002
000001
011012
140200
000006
000010
000012
000016
000022
000026
000032
000036
000042
000046
000052
000056
000062
000066
000072
000076
000102
000106
000112
000114
000120
000124
000130
000134
000140
000144
000150
000154
000160
000164

D.DRV= 0.
 D.UNIT = D.DRV+2
 DT.UNT= 000077
 DT.AVL= BIT15
 D.PRM = D.UNIT+2
 D.IW =BIT14
 D.DCY =BIT13
 D.ECC =BIT12
 D.RO =BIT11
 D.WO =BIT10
 D.RET =BIT9
 D.CYL =BIT8
 D.SEQ =BIT6
 D.BE =BIT5
 D.TR =BIT4
 D.WC =BIT3
 D.WCA =BIT2
 D.DC =BIT1
 D.DCA =BIT0
 DDEF=D.ECC+D.WC+D.DC+D.RET
 D.ZERO=BIT15+BIT7+D.IW
 D.PAT = D.PRM+2
 D.BB = D.PAT+2
 D.BB01 = D.BB+2
 D.BB02 = D.BB01+4
 D.BB03 = D.BB02+4
 D.BB04 = D.BB03+4
 D.BB05 = D.BB04+4
 D.BB06 = D.BB05+4
 D.BB07 = D.BB06+4
 D.BB08 = D.BB07+4
 D.BB09 = D.BB08+4
 D.BB10 = D.BB09+4
 D.BB11 = D.BB10+4
 D.BB12 = D.BB11+4
 D.BB13 = D.BB12+4
 D.BB14 = D.BB13+4
 D.BB15 = D.BB14+4
 D.BB16 = D.BB15+4
 D.BEC = D.BB16+4
 D.BGN1 = D.BEC+2
 D.END1 = D.BGN1+4
 D.BGN2 = D.END1+4
 D.END2 = D.BGN2+4
 D.BGN3 = D.END2+4
 D.END3 = D.BGN3+4
 D.BGN4 = D.END3+4
 D.END4 = D.BGN4+4
 D.BCYL = D.END4+4
 D.ECYL = D.BCYL+4
 D.XFRW = D.ECYL+4

:DRIVE NUMBER

: LOGICAL UNIT NUMBER OF DRIVE
 : SET WHEN NOT AVAILABLE FOR TESTING
 :HARDWARE QUESTION FLAGS
 :INITIAL WRITE
 :DIAGNOSTIC CYLINDERS
 :ECC CORRECTION ENABLED
 :READ ONLY
 :WRITE ONLY
 :RETRIES ENABLED
 :START/END CYLINDERS SPECIFIED
 :SEQUENTIAL ACCESS
 :BEGIN/END BLOCKS USED
 :WHEN D.BE=0: 1 - TRACKS, 0 - GROUPS
 :WRITE CHECKS ENABLED
 :ALWAYS WRITE CHECK
 :DATA COMPARES ENABLED
 :ALWAYS DATA COMPARE
 :DEFAULT D.PRM
 :BITS TO BE CLEARED
 :DATA PATTERN NUMBER
 :BAD BLOCK COUNT
 :BAD BLOCK 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 :BEGIN/END SET COUNT
 :BEGIN BLOCK 1
 :END
 :BEGIN BLOCK 2
 :END
 :BEGIN BLOCK 3
 :END
 :BEGIN BLOCK 4
 :END
 :BEGIN CYLINDER
 :END CYLINDER
 :MEGABITS WRITTEN COUNT

58	000166	D.XFRR =	D.XFRW+2	:MEGABITS READ COUNT
59	000170	D.HERR =	D.XFRR+2	:HARD ERROR COUNTER
60	000172	D.SERR =	D.HERR+2	:SOFT ERROR COUNTER
61	000174	D.SEEK =	D.SERR+2	: NUMBER OF SEEKS X1000
62	000176	D.ECCC =	D.SEEK+2	:ECC COUNTER
63	000200	D.SERN =	D.ECCC+2	:DRIVE SERIAL NUMBER
64				
65	000206	D.SIZE =	D.SERN+6	:SIZE OF DRIVE TABLE IN BYTES

```
1          ;KT MEMORY MANAGEMENT REGISTERS
2
3          172340      PAR0= 172340      ;KERNAL PAGE ADDRESS REGISTERS
4          172342      PAR1= 172342
5          172344      PAR2= 172344
6          172346      PAR3= 172346
7          172350      PAR4= 172350
8          172352      PAR5= 172352
9          172354      PAR6= 172354
10         172356      PAR7= 172356
11
12         172300      PDR0= 172300      ;KERNAL PAGE DESCRIPTOR REGISTERS
13         172302      PDR1= 172302
14         172304      PDR2= 172304
15         172306      PDR3= 172306
16         172310      PDR4= 172310
17         172312      PDR5= 172312
18         172314      PDR6= 172314
19         172316      PDR7= 172316
20         077400      MM.PLF= 077400      ;PAGE LENGTH FIELD
21                                     ;NORMALLY ALL SET
22         000100      MM.W= 000100      ;WRITTEN INTO (READ ONLY)
23         000010      MM.ED= 000010     ;EXPANSION DIRECTION
24                                     ;NORMALLY NOT SET
25         000006      MM.ACF= 000006    ;ACCESS CONTROL FIELD
26                                     ;NORMALLY ALL SET
27
28         177572      SR0= 177572      ;STATUS REGISTER 0
29         000001      MM.EN= 000001     ;ENABLE MANAGEMENT
30
31
32         ;DM PROGRAM HEADER DEFINITIONS
33
34         000000      DMTRLN= 0          ;OFFSET TO SIZE OF PROGRAM NEEDING DOWNLINE LOAD
35         000004      DMOVRL= 4         ;OFFSET TO SIZE OF OVERLAY
36         000040      DMMAIN= 40        ;OFFSET TO FIRST WORD OF MAIN PROGRAM
37         001000      DMFRST= 1000      ;ADDRESS IN DM FILE CONTAINING FIRST BYTE OF HEADER
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```
;USEFUL INSTRUCTION DEFINITIONS
.MACRO AND ARG,ADR                ;LOGICAL AND INSTRUCTION
.LIST
    BIC #^C<ARG>,ADR
.NLIST
.ENDM

.MACRO OR ARG,ADR                 ;LOGICAL OR INSTRUCTION
.LIST
    BIS #ARG,ADR
.NLIST
.ENDM

.MACRO PUSH ARG                   ;PUSH INSTRUCTION
.IRP X,<ARG>
.LIST
    MOV X,-(SP)
.NLIST
.ENDM
.ENDM

.MACRO POP ARG                     ;POP INSTRUCTION
.IRP X,<ARG>
.LIST
    MOV (SP)+,X
.NLIST
.ENDM
.ENDM

.MACRO .BR ADR                     ;A BRANCH TO THE NEXT LOCATION
.IF P2
    .IF ADR NE .
        .ERROR ;ILLEGAL .BR TO ADR
    .ENDC
.ENDC
.ENDM

.MACRO ASSUME FIRST CONDITION SECOND
.IF CONDITION <FIRST>-<SECOND>
.IFF
    .ERROR ;BAD ASSUME OF <FIRST> CONDITION <SECOND>
.ENDC
.ENDM
```

```

1          .SBTTL GLOBAL DATA SECTION
2
3          :++
4          : THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
5          : IN MORE THAN ONE TEST.
6          :--
7
20
21 002162          ERRTBLS          L$ERRTBLS:
    002162          ERRRTYP: .WORD 0
    002164          ERRNBR: .WORD 0
    002166          ERRMSG: .WORD 0
    002170          ERRBLK: .WORD 0
22
23 002172          000          000          ERRCHR: .BYTE 0,0          :FIRST BYTE LOADED WITH OUTPUT CHARACTER
24                                     :SECOND BYTE REMAINS ZERO TO STOP OUTPUT
25 002174          FFREE: .BLKW 1          :FIRST FREE WORD IN MEMORY
26 002176          FSIZE: .BLKW 1          :SIZE OF FREE MEMORY IN WORDS
27 002200          FMEM: .BLKW 1          :COPY OF FFREE AT END OF INIT SECTION
28 002202          FMEMS: .BLKW 1          :COPY OF FSIZE AT END OF INIT SECTION
29 002204          CTABS: .BLKW 1          :START OF CONTROLLER TABLE STORAGE
30 002206          CTRLRS: .BLKW 1          :COUNT OF UDA CONTROLLERS IN PTABLES
31 002210          TSTTAB: .BLKW 1          :POINTER TO FIRST CONTROLLER TABLE UNDER TEST
32 002212          DMPROG: .BLKW 1          :START ADDRESS OF DM PROGRAM
33 002214          DMEND: .BLKW 1          :END ADDRESS OF DM PROGRAM(FIRST FREE MEMORY ADR)
34 002216          DMENDS: .BLKW 1          :FREE MEMORY SIZE FROM END OF DM PROGRAM
35
36 002220          KTBASA: .BLKW 1          :HIGH TWO BYTES OF BASE ADDRESS FOR KT ACCESS
37 002222          KTBASO: .BLKW 1          :LOW BYTE OF ADDRESS FOR KT ACCESS
38
39 002224          IFLAGS: .BLKW 1          :FLAGS FROM INIT CODE FOR TEST 4
40          000002          ICONT ==BIT1          : CONTINUE EVENT FLAG
41          000004          IREST ==BIT2          : RESTART FLAG
42          000010          ISTRT ==BIT3          : START FLAG
43          000020          ISTRTH==BIT4          : START FLAG HOLD FOR T4UPRM ROUTINE
44 002226          TNUM: .BLKW 1          :NUMBER OF TEST EXECUTING
45 002230          URUN: .BLKW 1          :NUMBER OF UNITS TO RUN AT ONE TIME
46 002232          URNING: .BLKW 1          :NUMBER OF UNITS STILL RUNNING
47 002234          UCNT: .BLKW 1          :COUNTER OF UNITS UNDER TEST
48 002236          INTRCV: .BLKW 1          :INTERRUPT RECEIVED FLAG FOR INT TESTING
49
50 002240          FNAME:          :NAME OF DATA FILE
53 002240          132          125          104          .ASCIZ\ZUDDAO.PAK\
55          .EVEN
56 002254          000000          FDATA: .WORD 0
57 002256          000000          FILOPN: .WORD 0          :FILE OPEN WHEN NON-ZERO
58 002260          TEMP: .BLKW 12.          :TEMPORARY STORAGE FOR GMANI RESPONSES
59
60 002310          000001          PAT16C: .WORD 1          :COUNT OF WORDS IN DATA PATTERN 16
61 002312          000000          PAT16W: .WORD 0          :WORD SEQUENCE FOR DATA PATTERN 16
62 002314          000000          .WORD 0
63 002316          000000          .WORD 0
64 002320          000000          .WORD 0
65 002322          000000          .WORD 0
66 002324          000000          .WORD 0
67 002326          000000          .WORD 0
    
```

GLOBAL DATA SECTION

68	002330	000000	.WORD 0
69	002332	000000	.WORD 0
70	002334	000000	.WORD 0
71	002336	000000	.WORD 0
72	002340	000000	.WORD 0
73	002342	000000	.WORD 0
74	002344	000000	.WORD 0
75	002346	000000	.WORD 0
76	002350	000000	.WORD 0

```
1          ;CLOCK CONTROL
2
3 002352 000000      KW.CSR: .WORD 0          ;CSR OF CLOCK
4 002354 000000      KW.BRL: .WORD 0          ;BR LEVEL
5 002356 000000      KW.VEC: .WORD 0          ;VECTOR
6 002360 000000      KW.HZ: .WORD 0          ;HERTZ (50. OR 60.)
7 002362              KW.EL: .BLKW 2          ;ELAPSED TIME
8
9 002366 000000      NXMAD: .WORD 0          ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
10 002370 000000     KTMEM: .WORD 0          ;SET TO ALL ONES IF NO KT EXISTS
11
12 002372              STIME: .BLKW 2          ;STATISTICAL REPORT TIMER
13
14 002376              T2WRR: .BLKW 1          ;WRITE/READ REGION
15 002400              T2WRO: .BLKW 1          ;WRITE/READ OFFSET
16 002402              T2DR: .BLKW 1          ;DIAGNOSE REGION
17
18
19          ;ERROR LOG CONTROL WORDS
20
21 002404              LBUFS: .BLKW 1          ;START ADDRESS OF LOG/ZERO IF NONE
22 002406              LBUFN: .BLKW 1          ;ADDRESS FOR MORE DATA FOR LOG
23 002410              LBUFE: .BLKW 1          ;LAST ADDRESS AVAILABLE FOR LOG DATA
24
25          ;DISK DIAGNOSTIC DLL CONTROL WORDS
26
27 002412              DLL: .BLKW 1           ;DOWNLINE LOAD RESPONSE CODE = 0 - NO DATA,
28                                     ;1 - PROGRAM PROVIDED, 2- PROGRAM NOT FOUND
29 002414              DLLDR: .BLKW 1         ;DRIVE NUMBER REQUESTING PROGRAM
30 002416              DLLV: .BLKW 1         ;A VALUE FROM DM PROGRAM TO BE RETURNED
31 002420              DLLR: .BLKW 1         ;REGION
32 002422              DLLADR: .BLKW 2        ;ADDRESS WHERE PROGRAM STORED
33 002426              DLLSIZ: .BLKW 1       ;SIZE OF PROGRAM IN BYTES
34 002430              DLLNAM: .BLKW 2       ;NAME OF PROGRAM IN RAD50
```



```
1          .SBTTL GLOBAL TEXT SECTION
2
3
4          :++
5          : THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
6          : MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
7          : MORE THAN ONE TEST.
8          :--
9
10         :
11         : NAMES OF DEVICES SUPPORTED BY PROGRAM
12         :
13         :   DEVTYP <LOGICAL DISK DRIVE>
14         :
15         :   LSDVTYP::
16         :   .ASCIZ /LOGICAL DISK DRIVE/
17         :   .EVEN
18
19
20         : TEST DESCRIPTION
21         :
22         :   DESCRIPT <UDA-50 CONTROLLER AND DISK DRIVE DIAGNOSTIC>
23         :
24         :   L$DESC::
25         :   .ASCIZ /UDA-50 CONTROLLER A
26         :   .EVEN
27
28         :
29         : DESCRIPTIONS OF INDIVIDUAL TESTS
30         :
31         :   T1NAME: .ASCIZ\UNIBUS ADDRESSING\
32         :   T2NAME: .ASCIZ\DISK RESIDENT\
33         :   T3NAME: .ASCIZ\DISK FUNCTION\
34         :   T4NAME: .ASCIZ\DISK EXERCISER\
35
```

12	002434			
	002434	114	117	107
	002434			
22	002460			
	002460	125	104	101
	002460			
32	002534	125	116	111
33	002556	104	111	123
34	002574	104	111	123
35	002612	104	111	123

```

1          ;UNFORMATTED MESSAGES USED IN ERROR CALLS
2
3 002631    116    117    124  FMERRM: .ASCIZ\NOT ENOUGH MEMORY. SELECT FEWER UNITS TO TEST.\
4 002710    127    122    117  RSPPKE: .ASCIZ\WRONG OPCODE RECEIVED IN MESSAGE PACKET\
5 002760    115    105    123  RSPPRE: .ASCIZ\MESSAGE PACKET RECEIVED WITH WRONG REFERENCE NUMBER\
6 003044    115    105    123  RSPPNE: .ASCIZ\MESSAGE PACKET RECEIVED WITH UNKNOWN REQUEST NUMBER\
7 003130    105    122    122  RSPSTE: .ASCIZ\ERROR STATUS CODE REPORTED IN MESSAGE COMMAND\
8 003206    125    104    101  LOADM1: .ASCIZ\UDA RETURNED ERROR TO SEND STUD DATA COMMAND\
9 003263    125    104    101  INTHD: .ASCIZ\UDA INITIALIZE ERROR\
10 003310   105    122    122  RWRDEM: .ASCIZ\ERROR READING 'DIAGNOSTIC MACHINE' PROGRAM FILE\
11 003370   124    127    117  MLDREM: .ASCIZ\TWO P-TABLES POINT TO SAME DRIVE\
12 003431   115    117    122  TOOMEM: .ASCIZ\MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER\
13 003513   124    101    102  BADT: .ASCIZ\TABLE INCONSISTANCY ERROR. START PROGRAM AGAIN.\
14 003573   116    125    115  T4BB: .ASCIZ\NUMBER OF BAD BLOCKS\
15 003620   102    101    104  T4BBI: .ASCIZ\BAD BLOCK\
16 003632   122    105    101  T4RO: .ASCIZ\READ ONLY\
17 003644   127    122    111  T4WO: .ASCIZ\WRITE ONLY\
18 003657   103    110    105  T4WCA: .ASCIZ\CHECK ALL WRITES BY READING\
19 003713   122    101    116  T4WCR: .ASCIZ\RANDOMLY CHECK WRITES BY READING\
20 003754   104    101    124  T4DP: .ASCIZ\DATA PATTERN - 0 FOR RANDOM SELECTION\
21 004022   105    116    101  T4ECC: .ASCIZ\ENABLE ECC DATA CORRECTION\
22 004055   103    117    115  T4DCA: .ASCIZ\COMPARE ALL DATA READ\
23 004103   122    101    116  T4DCR: .ASCIZ\RANDOMLY COMPARE DATA READ\
24 004136   105    116    101  T4RET: .ASCIZ\ENABLE RETRIES\
25 004155   122    101    116  T4SEK: .ASCIZ\RANDOM SEEK MODE\
26 004176   040    040    000  T4OPT7: .ASCIZ\ \
27 004201   116    125    115  T4BE: .ASCIZ\NUMBER OF BEGIN/END SETS\
28 004232   102    105    107  T4BEG: .ASCIZ\BEGIN BLOCK\
29 004246   105    116    104  T4END: .ASCIZ\END BLOCK\
30 004260   116    125    115  T4TRC: .ASCIZ\NUMBER OF TRACKS TO TEST\
31 004311   124    122    101  T4TRAK: .ASCIZ\TRACK\
32 004317   116    125    115  T4GRC: .ASCIZ\NUMBER OF GROUPS TO TEST\
33 004350   107    122    117  T4GRP: .ASCIZ\GROUP\
34 004356   104    117    040  T4CYL: .ASCIZ\DO YOU WISH TO LIMIT THE CYLINDERS TESTED\
35 004430   123    124    101  T4CYLB: .ASCIZ\STARTING CYLINDER\
36 004452   105    116    104  T4CYLE: .ASCIZ\ENDING CYLINDER\
37 004472   116    125    115  T4DPC: .ASCIZ\NUMBER OF WORDS IN DATA PATTERN 16\
38 004535   104    101    124  T4DPD: .ASCIZ\DATA WORD\
39 004547   104    115    040  GTDRV1: .ASCIZ\DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE\
40 004622   125    104    101  LOOP00: .ASCIZ\UDASA NON-ZERO AFTER ENTRY INTO LOOP MODE OR WRITING UDASA\
41 004715   125    104    101  LOOP01: .ASCIZ\UDASA NEVER BECAME NON-ZERO AFTER WRITING WITH NON-ZERO DATA\
42 005012   104    101    124  LOOP02: .ASCIZ\DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST\
43 005101   111    116    103  INTST2: .ASCIZ\INCORRECT BR LEVEL\
44 005124   116    117    040  INTST4: .ASCIZ\NO INTERRUPTS EVER RECEIVED FROM UDA\
45 005171   101    122    105  INITWC: .ASCIZ\ARE YOU SURE CUSTOMER DATA CAN BE DESTROYED\
46 005245   106    101    111  DMNF: .ASCIZ\FAILED TO READ DM PROGRAM FROM DATA FILE\
47          .EVEN
    
```

```

1          ; FORMAT STATEMENTS USED IN PRINT CALLS
2
20 005316      045      124      000  ERRONE: .ASCIZ\%T\
21 005321      045      116      000  ERRNL:  .ASCIZ\%N\
22 005324      045      101      124  INTM1:  .ASCIZ\%ATIME-OUT ERROR WHILE WAITING FOR RESPONSE IN UDASA REGISTER%N\
23 005424      045      101      011  INTM2:  .ASCIZ\%A          UDASA REGISTER = %06%N\
24 005456      045      101      125  INTM3:  .ASCIZ\%AUDA RESIDENT DIAGNOSTICS DETECTED FAILURE%N\
25 005534      045      101      125  INTM4:  .ASCIZ\%AUDASA REGISTER DID NOT RETURN CORRECT VALUE%N\
26 005614      045      101      011  INTM5:  .ASCIZ\%A          EXPECTED = %06%N\
27 005640      045      101      011  INTM6:  .ASCIZ\%A          ACTUAL   = %06%N\
28 005664      045      101      122  INTB1M: .ASCIZ\%ARING BUFFER WAS NOT CLEARED BY UDA%N\
29 005733      045      101      103  INTB2M: .ASCIZ\%ACURRENT CONTENTS OF RING BUFFER:%N\
30 006000      045      123      066  INTB3M: .ASCIZ\%S6%AADDRESS%S4%ACONTENTS%N\
31 006034      045      117      061  INTB4M: .ASCIZ\%013%011%N\
32 006047      045      101      106  RWRDF1: .ASCIZ\%AFILE BEING READ IS '%T%A'%N\
33 006105      045      101      104  RWRDF2: .ASCIZ\%ADID NOT FIND START AND NULL FRAMES WHERE EXPECTED%N\
34 006173      045      101      103  RWRDF4: .ASCIZ\%ACHECKSUM ERROR%N\
35 006216      045      124      045  ERRMB:  .ASCIZ\%T%A  DM PC:%04%A  UDA AT %06%A  \
36 006260      045      101      104  ERRMBD: .ASCIZ\%ADRIVE %Z3%A  \
37 006300      045      101      122  ERRMRT: .ASCIZ\%ARUNTIME %D3%A:%Z2%A:%Z2%\
38 006332      045      101      040  ERRME1: .ASCIZ\%A * * * ERROR PROCESSING DM MESSAGE STRING * * *%N\
39 006425      045      101      122  MXFERP: .ASCIZ\%AREACHED TRANSFER LIMIT - TESTING STOPPED%N\
40 006502      045      116      045  ERR LIM: .ASCIZ\%N%AUNIT %Z2%A REACHED ERROR LIMIT - WILL NO LONGER BE TESTED.%N\
41 006603      045      101      040  LOOP03: .ASCII\%A  DATA SENT      : %07%N\
42 006636      045      101      040  .ASCIZ\%A  DATA RECEIVED : %07%N\
43 006672      045      116      045  INTST0: .ASCIZ\%N%A TESTING INTERRUPT ABILITY OF UDA AT ADR %06%A VEC %03%A... \
44 006771      045      101      103  INTST1: .ASCIZ\%ACOMPLETED%N\
45 007007      045      101      125  INTST3: .ASCIZ\%AUDA INTERRUPTED AT BR LEVEL %01%N\
46 007053      045      116      045  INITWA: .ASCIZ\%N%ACUSTOMER DATA WILL BE DESTROYED ON:%N%S5%AUNIT%S5%AUDA AT%S3%ADRIVE%N\
47 007165      045      123      066  INITWB: .ASCIZ\%S6%D2%S6%06%S4%D3%N\
48 007212      045      116      045  T4WARN: .ASCIZ\%N%AMANUAL INTERVENTION NOT ALLOWED. TEST 4 USING DEFAULT PARAMETERS%N\
49 007321      045      116      045  T4QHED: .ASCIZ\%N%ATHE FOLLOWING QUESTIONS REFER TO UNIT %Z2%A UDA AT %06%A DRIVE %Z3%N\
50 007432      045      116      045  T4OPT1: .ASCIZ\%N%ADO YOU WISH TO:%N\
51 007460      045      101      040  T4OPT2: .ASCIZ\%A  0 - TEST ENTIRE AREA SELECTED%N\
52 007525      045      101      040  T4OPT3: .ASCIZ\%A  1 - SPECIFY BEGIN/END SETS TO TEST%N\
53 007577      045      101      040  T4OPT4: .ASCIZ\%A  2 - SPECIFY TRACKS AND CYLINDERS TO TEST%N\
54 007657      045      101      040  T4OPT5: .ASCIZ\%A  3 - SPECIFY GROUPS AND CYLINDERS TO TEST%N\
55 007737      045      101      040  T4OPT6: .ASCIZ\%A  4 - SPECIFY CYLINDERS TO TEST\
56 010002      045      101      114  INP28A: .ASCIZ\%ALIMITS - LO= 0, HI= 268435455%N\
57 010044      045      101      111  INP28B: .ASCIZ\%AINVALID CHAR, TYPE DECIMAL NUMBER 0 TO 268435455%N\
58 010131      045      116      045  MESSG:  .ASCIZ\%N%AUNIT %Z2%A UDA AT %06%A DRIVE %Z3%S\
59 010201      045      101      104  CTABEM: .ASCIZ\%ADIFFERENT VECTORS, BR LEVELS OR BURST RATES SPECIFIED FOR UDA AT %06%N\
60 010312      045      116      045  T2WARN: .ASCIZ\%N%AMANUAL INTERVENTION NOT ALLOWED. TEST 2 RUNNING UNATTENDED.%N\
61 010414      045      116      045  T2CMS1: .ASCIZ\%N%A TEST #2 MANUAL INTERVENTION ON UNIT %Z2%A UDA AT %06%A DRIVE %Z3%N\
62 010523      045      101      124  T2CMS2: .ASCII\%ATO WRITE AND READ MEMORY:%N\
63 010560      045      101      040  .ASCII\%A  W DATA REGION OFFSET%N\
64 010612      045      101      040  .ASCIZ\%A  R REGION OFFSET%N\
65 010640      045      101      124  T2CMS3: .ASCII\%ATO RUN A DIAGNOSTIC:%N\
66 010670      045      101      040  .ASCIZ\%A  D REGION%N\
67 010707      045      101      124  T2CMS4: .ASCII\%ATO EXIT QUESTIONING:%N\
68 010737      045      101      040  .ASCII\%A  E%N\
69 010746      045      101      104  .ASCIZ\%ADATA, REGION AND OFFSET ARE HEX VALUES.%N\
70 011022      045      101      077  T2CMS5: .ASCIZ\%A? INPUT ERROR%N\
71 011044      045      101      116  NOCLOCK: .ASCIZ\%ANO LINE CLOCK AVAILABLE FOR TIMING EVENTS%N\
72 011122      045      101      116  RSPTOM: .ASCIZ\%ANO INTERRUPT FROM UDA AT %06%A FOR 3-MINUTES%N\
73 011203      045      101      106  RSPTMM: .ASCIZ\%AFATAL ERROR FROM UDA AT %06%A  UDASA CONTAINS %06%N\
74 011274      045      101      115  INITMM: .ASCIZ\%AMEMORY ACCESS ERROR TO UDA REGISTERS AT %06%N\
    
```

GLOBAL TEXT SECTION

75	011354	045	101	125	INTM7:	.ASCIZ\%AUDASA DID NOT GO TO ZERO AFTER STEP 3 WRITE WITH PURGE/POLL BIT SET%\
76	011464	040	040	040	INTM8:	.ASCIZ\ UDASA CONTAINS %06%\
77	011514	045	101	105	LOGDAT:	.ASCIZ\%AERROR - DATA PLACED IN LOG BUFFER%\
78	011562	045	101	105	LOGFUL:	.ASCIZ\%AERROR - DATA LOST BECAUSE LOG BUFFER FULL%\
79	011640	045	116	045	LOGM1:	.ASCIZ\%N%CONTENTS OF ERROR LOG:%%\
80	011675	045	116	045	LOGM2:	.ASCIZ\%N%END OF ERROR LOG%\
81	011724	045	116	045	LOGM3:	.ASCIZ\%N%AERROR LOG IS EMPTY%\
82						.EVEN

```

1          .SBTTL GLOBAL ERROR REPORT SECTION
2
3          :++
4          : THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
5          : USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PRINTB
6          : (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
7          :--
8          177777 SVCINS= -1          ; LIST INSTRUCTIONS, SHIFTED RIGHT
9          177777 SVCTST= -1         ; LIST TEST TAGS, SHIFTED RIGHT
10         177777 SVCSUB= -1        ; LIST SUBTEST TAGS, SHIFTED RIGHT
11         177777 SVCGBL= -1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
12         177777 SVCTAG= -1      ; LIST OTHER TAGS, SHIFTED RIGHT
13
14 011756 BGNMSG INTR1              ;TIME OUT ERROR
15 011756 PRINTB #INTM1
16 011776 PRINTB #INTM2,R2
17 012020 ENDMSG
18
19 012022 BGNMSG INTR2              ;UDA DIAGS FAILED
20 012022 PRINTB #INTM3
21 012042 PRINTB #INTM2,R2
22 012064 ENDMSG
23
24 012066 BGNMSG INTR3              ;IMPROPER RESPONSE
25 012066 PRINTB #INTM4
26 012106 PRINTB #INTM5,R1
27 012130 PRINTB #INTM6,R2
28 012152 ENDMSG
29
30 012154 BGNMSG INTR4              ;UDA NON-EXISTANT
31 012154 PRINTB #INITMM,(R5)
32 012176 ENDMSG
33
34 012200 BGNMSG INTR5
35 012200 PRINTB #INTM7
36 012220 PRINTX #INTM8,R2
37 012242 ENDMSG
38
39 012244 BGNMSG INTBF              ;RING BUFFER NOT CLEARED
40 012244 PRINTB #INTB1M
41 012264 PRINTX #INTB2M
42 012304 PRINTX #INTB3M
43 012324 013702 002174 MOV FFREE,R2
44 012330 010103 MOV R1,R3
45 012332 INTBFL: PRINTX #INTB4M,R2,(R2)
46 012356 062702 000002 ADD #2,R2
47 012362 005303 DEC R3
48 012364 003362 BGT INTBFL
49 012366 ENDMSG
50
51 012370 BGNMSG RWRDM1
52 012370 PRINTB #RWRDF1,#FNAME
53 012414 PRINTB #RWRDF2
54 012434 ENDMSG
55
56 012436 BGNMSG RWRDM3
57 012436 PRINTB #RWRDF1,#FNAME
    
```

```
58 012462          PRINTB #RWRDF4
59 012502          ENDMSG
60
61 012504          BGNMSG LOOP
62 012504          PRINTB #LOOP03,R2,2(R4)
63 012532          ENDMSG
64
65 012534          BGNMSG LOOPA
66 012534          PRINTB #LOOP03,#140000,2(R4)
67 012564          ENDMSG
68
69 012566          BGNMSG INTERR
70 012566          PRINTB #INTST3,R1
71 012610          ENDMSG
72
73 012612          BGNMSG CTABE
74 012612          PRINTB #CTABEM,(R3)
75 012634          ENDMSG
76
77 012636          BGNMSG RSPTOE
78 012636          PRINTB #RSPTOM,(R5)
79 012660          ENDMSG
80
81 012662          BGNMSG RSPTME
82 012662          PRINTB #RSPTMM,(R5),R1
83 012706          ENDMSG
```

```
84
85 000001
86 000001
87 000001
88 000001
89 000001
```

```
SVCINS= 1
SVCTST= 1
SVCSUB= 1
SVCGBL= 1
SVCTAG= 1
```

```
: LIST INSTRUCTIONS, SHIFTED RIGHT
: LIST TEST TAGS, SHIFTED RIGHT
: LIST SUBTEST TAGS, SHIFTED RIGHT
: LIST GLOBAL TAGS, SHIFTED RIGHT
: LIST OTHER TAGS, SHIFTED RIGHT
```

```
1      .SBTTL GLOBAL SUBROUTINES SECTION
2
3      ;MEMORY ALLOCATION ERROR
4      ;
5      ;THIS ROUTINE PRINTS A SYSTEM FATAL ERROR AND EXITS THE TEST
6
7      FMERR: ERRSF 1,FMERRM
8
9      DOCLN                                ;ABORT
10
11     TRAP C$ERSF
12     .WORD 1
13     .WORD FMERRM
14     .WORD 0
15
16     TRAP C$DCLN
17
18     012710 104454
19     012712 000001
20     012714 002631
21     012716 000000
22     012720 104444
```

```
1      ;ALOCM
2      ;
3      ;ALLOCATE A BLOCK OF FREE MEMORY.  REPORT ERROR IF MEMORY EXHAUSTED.
4      ;
5      ;INPUTS:
6      ;   R1 - NUMBER OF WORDS TO ALLOCATE
7      ;   FFREE - FIRST FREE WORD IN MEMORY
8      ;   FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS
9      ;OUTPUTS:
10     ;   R1 - ADDRESS OF FIRST WORD OF ALLOCATED MEMORY
11     ;   FFREE - NEW FIRST FREE WORD IN MEMORY
12     ;   FSIZE - SIZE OF FREE MEMORY LEFT AFTER ALLOCATION
13     ;SYSTEM FATAL ERROR WILL BE REPORTED IF NOT ENOUGH MEMORY AVAILABLE
14     ;AND ENTIRE PROGRAM WILL BE STOPPED.
15
16 012722      ALOCM:  PUSH FFREE                ;SAVE FFREE AT ENTRY
17 012722      013746      002174              MOV FFREE,-(SP)
18 012726      160137      002176              SUB R1,FSIZE                ;REDUCE SIZE OF FREE MEMORY
19 012732      002766                          BLT FMERR                  ;REPORT ERROR IF NOT ENOUGH MEMORY
20 012734      060101                          ADD R1,R1                  ;CHANGE WORDS TO BYTES
21 012736      060137      002174              ADD R1,FFREE              ;CALCULATE NEW START OF FREE MEMORY
22 012742      012742      012601              POP R1                    ;GET START OF ALLOCATED MEMORY
23 012744      000207                          MOV (SP)+,R1
24                                     RETURN
```



```
1      :HCOMM
2      :
3      :ALLOCATES MEMORY FOR HOST COMM AREA AND PACKET BUFFERS WITH ONE
4      :DESCRIPTOR IN EACH RING. TO BE CALLED AFTER INITIALIZING
5      :A CONTROLLER WITH SA.MSG=0 AND SA.CMD=0.
6      :
7      :INPUTS:
8      :      R5 - ADDRESS OF CONTROLLER TABLE
9      :
10     :OUTPUTS:
11     :      CONTROLLER TABLE POINTING TO HOST COMM AREA
12     :      RING POINTERS TO PACKETS
13     :      R4 - ADDRESS OF HOST COMM AREA
14 012746 012701 000166      HCOMM: MOV #HC.SIZ/2,R1      ;GET SIZE OF AREA TO ALLOCATE
15 012752 004737 012722      CALL ALOCM          ;ALLOCATE THE MEMORY
16 012756 010104              MOV R1,R4           ;GET ADDRESS OF HOST COMM AREA
17 012760 010465 000016      MOV R4,C.RING(R5)  ;PLACE IN CONTROLLER TABLE
18 012764 062701 000020      ADD #HC.MPK,R1     ;COMPUTE START OF MESSAGE PACKET
19 012770 010164 000004      MOV R1,HC.MSG(R4) ;PLACE IN RING
20 012774 062701 000064      ADD #<HC.CPK-HC.MPK>,R1 ;COMPUTE START OF COMMAND PACKET
21 013000 010164 000010      MOV R1,HC.CMD(R4) ;PLACE IN RING
22 013004 000207              RETURN
```

```

1      :TINIT
2      :
3      :INITIALIZE VARIABLES FOR TEST
4      :
5      :INPUTS:
6      :      R1 - TEST NUMBER
7      :
8      :OUTPUTS:
9      :      LBUFS - CLEARED (DELETES ERROR LOG)
10     :      FFREE - FROM FMEM
11     :      FSIZE - FROM FMEMS
12     :      TNUM - TEST NUMBER FROM R1
13     :      ALL REGISTERS CLOBERED
14 013006 004737 024414      TINIT: CALL RESET           ;RESET ALL DEVICES
15 013012 005037 002404      CLR LBUFS           ;CLEAR ERROR LOG BUFFER POINTER
16 013016 013737 002200 002174  MOV FMEM,FFREE       ;INIT FFREE
17 013024 013737 002202 002176  MOV FMEMS,FSIZE     ;INIT FSIZE
18 013032 020137 002226      CMP R1,TNUM         ;SEE IF SAME TEST RUNNING
19 013036 001007              BNE TINITR           ;IF NOT, GO TO READ DM PROGRAM
20 013040 013737 002214 002174  MOV DMEND,FFREE     ;CHANGE FREE MEMORY TO LEAVE
21 013046 013737 002216 002176  MOV DMENDS,FSIZE    ; DM PROGRAM ALLOCATED
22 013054 000207              RETURN
23
24 013056              TINITR:           ;BRESET ;RESET ALL UNITS
25 013056 004737 013070      CALL READDM        ;READ DM PROGRAM
26 013062 010137 002226      MOV R1,TNUM         ;STORE TEST NUMBER TO SHOW DM PROGRAM IN MEMORY
27 013066 000207              RETURN
    
```

```

1      :READDM
2
3      :READ A DM PROGRAM INTO FREE MEMORY
4
5      :INPUTS:
6      R1 - TEST NUMBER
7
8      :OUTPUTS:
9      DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
10     R1 - UNCHANGED
11     CARRY CLEAR IF NO ERROR, CARRY SET IF PROGRAM NOT FOUND
12     :ALL REGISTERS BUT R1 ARE USED AND PREVIOUS CONTENTS DESTROYED
13 013070 013737 002174 002212 READDM: MOV FFREE,DMPROG          ;GET STORAGE ADDRESS
14 013076 004737 023734          CALL RDREC
15 013102 103407          BCS README          ;CHECK IF ERROR
16 013104 013737 002174 002214 MOV FFREE,DMEND      ;SAVE END OF ADDRESS OF DM PROGRAM
17 013112 013737 002176 002216 MOV FSIZE,DMENDS    ; AND CURRENT SIZE OF FREE MEMORY
18 013120 000207          RETURN
19
20 013122          README: ERRSF 2,DMNF          ;REPORT DM PROGRAM NOT FOUND
    013122 104454          TRAP C$ERSF
    013124 000002          .WORD 2
    013126 005245          .WORD DMNF
    013130 000000          .WORD 0
21 013132          DOCLN
    013132 104444          TRAP C$DCLN
    
```

```

1      ;RUNDM
2      ;
3      ;LOAD AND RUN A DM PROGRAM IN THE CONTROLLERS. RETURN WHEN ALL
4      ;DM PROGRAMS HAVE TERMINATED.
5      ;
6      ;INPUTS:
7      ;   TSTTAB - POINTER TO FIRST CONTROLLER TABLE
8      ;   R1 - NUMBER OF CONTROLLERS TO TEST
9      ;IMPLICIT INPUTS:
10     ;   DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
11     ;OUTPUTS:
12     ;   Z SET IF NO CONTROLLERS SUCCESSFULLY STARTED
13     ;ALL REGISTERS ARE USED AND PREVIOUS CONTENTS DESTROYED.
14
15 013134 010137 002230  RUNDM:  MOV R1,URUN          ;SAVE NUMBER OF UNITS TO RUN
16 013140 005037 002232          CLR URNING        ;CLEAR NUMBER OF UNITS RUNNING
17
18     ;LOAD DM PROGRAM INTO EACH CONTROLLER
19
20 013144 013737 002230 002234  MOV URUN,UCNT      ;SET COUNTER OF UNITS
21 013152 013705 002210          MOV TSTTAB,R5     ;GET FIRST CONTROLLER TABLE
22 013156 005065 000014  LDDM:  CLR C.FLG(R5) ;CLEAR ALL FLAGS
23 013162 016537 000002 002074  MOV C.UNIT(R5),L$LUN ;SEE IF UNIT TO BE TESTED
24 013170 100405          BMI LDNEXT          ;IF NOT, DON'T LOAD THIS UNIT
25 013172          ASSUME CT.AVL EQ BIT15
26 013172 004737 021204          CALL LOADDM      ;LOAD THE DM PROGRAM
27 013176 001402          BEQ LDNEXT          ;IF ERROR, GO TO NEXT CONTROLLER
28 013200 005237 002232          INC URNING      ;IF NO ERROR, COUNT UNIT RUNNING
29 013204 062705 000046  LDNEXT: ADD #C.SIZE,R5 ;MOVE TO NEXT CONTROLLER TABLE
30 013210 005337 002234          DEC UCNT         ;CHECK IF MORE CONTROLLERS
31 013214 001360          BNE LDDM          ;LOAD NEXT
32
33     ;CHECK IF ANY CONTROLLERS LOADED
34
35 013216 005737 002232          TST URNING        ;ANY UNITS LOADED?
36
37     ;THE DM PROGRAMS ARE NOW IN CONTROL
38     ;RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
39
40 013222 000207          RETURN
    
```

```

1      ;RESPDM
2
3      ;
4      ;RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
5      ;HAVE TERMINATED.
6 013224 013705 002210      RESPDM: MOV TSTTAB,R5      ;GET CONTROLLER TABLE ADDRESS
7 013230 013737 002230 002234      MOV URUN,UCNT      ;SET COUNTER OF UNITS
8 013236 016504 000016      RESPCT: MOV C.RING(R5),R4      ;GET HOST COMM AREA ADDRESS
9 013242 032765 000002 000014      BIT #CT.RN,C.FLG(R5)      ;CHECK IF PROGRAM RUNNING
10 013250 001446      BEQ RSPNXT      ;IF NOT, LOOK AT NEXT
11 013252 016537 000002 002074      MOV C.UNIT(R5),L$LUN      ;STORE UNIT NUMBER UNDER TEST
12 013260 032765 000010 000014      BIT #CT.MSG,C.FLG(R5)      ;SEE IF INTERRUPT RECEIVED
13 013266 001077      BNE RSPIN      ;IF SO, LOOK AT PACKET
14 013270 032765 000004 000014      BIT #CT.CMD,C.FLG(R5)      ;SEE IF COMMAND HAS BEEN SENT
15 013276 001540      BEQ RSPOU      ;IF NOT, SEND ONE
16
17      ;CHECK IF UDA STILL RUNNING
18
19 013300 011503      MOV (R5),R3      ;GET ADDRESS OF UDAIP
20 013302 016301 000002      MOV 2(R3),R1      ;LOOK AT UDASA REGISTER
21 013306 001405      BEQ RSPTM      ;IF ZERO, UDA STILL RUNNING
22 013310      ERRDF 35,0,RSPTME      ;REPORT UDA HAS FATAL ERROR
23 013310 104455      TRAP C$ERDF
24 013312 000043      .WORD 35
25 013314 000000      .WORD 0
26 013316 012662      .WORD RSPTME
27 013320 000453      BR RSPDRP      ;DROP CONTROLLER FROM TESTING
28
29      ;CHECK FOR TIMEOUT OF RESPONSE
30
31 013322      RSPTM:
32 013322 005737 002352      TST KW.CSR      ;SEE IF A CLOCK ON SYSTEM
33 013326 001416      BEQ RSPNTO      ;DON'T TIME IF NO CLOCK
34 013330 023765 002364 000042      CMP KW.EL+2,C.TOH(R5)      ;COMPARE TO TIMEOUT COUNTER
35 013336 101005      BHI RSPTMO
36 013340 001011      BNE RSPNTO
37 013342 023765 002362 000040      CMP KW.EL,C.TO(R5)
38 013350 103405      BLO RSPNTO      ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
39 013352      RSPTMO: ERRDF 36,0,RSPTOE      ;REPORT TIMEOUT ERROR
40 013352 104455      TRAP C$ERDF
41 013354 000044      .WORD 36
42 013356 000000      .WORD 0
43 013360 012636      .WORD RSPTOE
44 013362 000432      BR RSPDRP      ;DROP CONTROLLER FROM TESTING
45 013364      RSPNTO:
46 013364      BREAK      ;ALLOW DRS TO SEE TERMINAL INPUT
47 013364 104422      TRAP C$BRK
48
49      ;CHECK FOR TIME TO PRINT STATISTICAL REPORT
50
51 013366 005737 002352      RSPNXT: TST KW.CSR      ;ANY CLOCK ON SYSTEM?
52 013372 001420      BEQ RSPNRP      ;BYPASS IF NOT
53 013374 023737 002364 002374      CMP KW.EL+2,STIME+2      ; A STATISTICAL REPORT
54 013402 101005      BHI RSPRPT
55 013404 001013      BNE RSPNRP
56 013406 023737 002362 002372      CMP KW.EL,STIME
57 013414 103407      BLO RSPNRP
    
```

```
54 013416          RSPRPT: DORPT          ;PRINT THE REPORT
    013416 104424          ;                               TRAP C$DRPT
55 013420 012700 001604          MOV #15.*60.,R0          ;SET TIME FOR NEXT REPORT
56 013424 012701 002372          MOV #STIME,R1          ;AT 15 MINUTES FROM NOW
57 013430 004737 022660          CALL SETTO
58
59          ;SWITCH TO NEXT CONTROLLER
60
61 013434 062705 000046          RSPNRP: ADD #C.SIZE,R5          ;MOVE TO NEXT TABLE
62 013440 005337 002234          DEC UCNT          ;CHECK IF MORE CONTROLLERS
63 013444 001274          BNE RESPCT          ;LOOK AT NEXT CONTROLLER
64 013446 000666          BR RESPDM          ;LOOK AT FIRST CONTROLLER AGAIN
65
66          ;REMOVE A CONTROLLER FROM TESTING
67
68 013450 042765 000012 000014 RSPDRP: BIC #CT.RN+CT.MSG,C.FLG(R5) ;CLEAR PROGRAM RUNNING
69 013456 005337 002232          DEC URNING          ;REDUCE RUNNING CONTROLLERS COUNT
70 013462 001341          BNE RSPNXT          ;IF ANY STILL RUNNING, LOOK AT THEM
71 013464 000207          RETURN          ;ELSE RETURN TO TEST SECTION
```

```

1          ;CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
2
3          ;CHECK FOR PROPER OPCODE IN END PACKET
4
5 013466 012700 000204          RSPIN: MOV #OP.END+OP.SSD,R0          ;GET SEND DATA END PACKET OPCODE
6 013472 032765 000020 000014  BIT #CT.REQ,C.FLG(R5)          ;LOOK IF SEND DATA OR RECEIVE DATA
7 013500 001402          BEQ RSPMWR
8 013502 012700 000205          MOV #OP.END+OP.RSD,R0          ;CHANGE TO RECEIVE DATA END PACKET OPCODE
9 013506 120064 000030  RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)          ;COMPARE TO OPCODE IN END PACKET
10 013512 001405          BEQ RSPSTS
11 013514          ERRHRD 25,RSPPKC          ;REPORT OPCODE ERROR IN RESPONSE PACKET
    013514 104456          TRAP C$ERHRD
    013516 000031          .WORD 25
    013520 002710          .WORD RSPPKC
    013522 000000          .WORD 0
12 013524 000751          BR RSPDRP          ;DROP UNIT FROM TESTING
13
14          ;LOOK AT STATUS CODE
15
16 013526 032764 000037 000032  RSPSTS: BIT #ST.MSK,HC.MPK+P.STS(R4)          ;CHECK FOR STATUS CODE ST.SUC (ZERO)
17 013534 001405          BEQ RSPREF
18 013536          ERRHRD 26,RSPSTE          ;REPORT ERROR STATUS CODE
    013536 104456          TRAP C$ERHRD
    013540 000032          .WORD 26
    013542 003130          .WORD RSPSTE
    013544 000000          .WORD 0
19 013546 000740          BR RSPDRP          ;DROP UNIT FROM TESTING
20
21          ;CHECK FOR EXPECTED REFERENCE NUMBER
22
23 013550 026564 000044 000020  RSPREF: CMP C.REF(R5),HC.MPK+P.CRF(R4)          ;CHECK IF CORRECT REF NUMBER
24 013556 001405          BEQ RSPPTW
25 013560          ERRHRD 31,RSPPRE
    013560 104456          TRAP C$ERHRD
    013562 000037          .WORD 31
    013564 002760          .WORD RSPPRE
    013566 000000          .WORD 0
26 013570 000727          BR RSPDRP          ;DROP UNIT FROM TESTING
27
28          ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
29
30 013572 032765 000020 000014  RSPPTW: BIT #CT.REQ,C.FLG(R5)          ;CHECK IF RESPONSE FROM DM PROGRAM
31 013600 001433  RSPOU: BEQ RSPOUT          ;LOOK AT REQUEST NUMBER IF SO
    
```

```

1          ;MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
2
3 013602 016401 000260      RSPPT2: MOV HC.BF2(R4),R1          ;GET REQUEST NUMBER
4 013606 020127 000017      CMP R1,#DSPSIZ          ;CHECK IF IN EXPECTED RANGE
5 013612 103405              BLO RSPPT3
6 013614              ERRHRD 32,RSPNE          ;BAD REQUEST NUMBER
   013614 104456              TRAP          C$ERHRD
   013616 000040              .WORD          32
   013620 003044              .WORD          RSPNE
   013622 000000              .WORD          0
7 013624 000711              BR RSPDRP          ;DROP UNIT FROM TESTING
8
9 013626 012700 000004      RSPPT3: MOV #OP.SSD,R0          ;BUILD A SEND DATA COMMAND PACKET
10 013632 004737 021624     CALL BLDCMD          ; FOR ANSWER TO DM PROGRAM
11 013636 012700 000164     MOV #HC.BF1,R0          ;POINT TO BUFFER IN PACKET
12 013642 004737 022006     CALL CLRBUF          ; AND CLEAR BUFFER
13 013646 010403              MOV R4,R3          ;R3 POINTS TO COMMAND BUFFER
14 013650 062704 000074     ADD #HC.BSZ,R4          ;R4 POINTS TO MESSAGE BUFFER
15 013654 011401              MOV (R4),R1          ;GET REQUEST NUMBER
16 013656 012423              MOV (R4)+,(R3)+      ;PUT REQUEST NUMBER INTO COMMAND PACKET
17 013660 060101              ADD R1,R1          ;DOUBLE REQUEST NUMBER
18 013662 004771 013772     CALL @RSPDSP(R1)      ;CALL REQUESTED ROUTINE
19 013666 001270              BNE RSPDRP          ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
20
21
22          ;SEND COMMAND BACK TO UDA
23
24 013670 042765 000010 000014  RSPOUT: BIC #CT.MSG,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
25 013676 032765 000020 000014  BIT #CT.REQ,C.FLG(R5)          ;CHECK WHICH COMMAND TO SEND
26 013704 001014              BNE RSPOU2          ;BRANCH IF RESPONSE TO REQUEST
27
28 013706 012700 000005              MOV #GP.RSD,R0          ;BUILD RECEIVE DATA COMMAND
29 013712 004737 021624     CALL BLDCMD
30 013716 012700 000260     MOV #HC.BF2,R0          ;POINT TO MESSAGE BUFFER
31 013722 004737 022006     CALL CLRBUF          ; AND CLEAR IT
32 013726 052765 000020 000014  BIS #CT.REQ,C.FLG(R5)      ;SET REQUEST BIT
33 013734 000403              BR RSPOU3
34
35 013736 042765 000020 000014  RSPOU2: BIC #CT.REQ,C.FLG(R5)      ;CLEAR REQUEST BIT
36 013744              RSPOU3:
37 013744 004737 021710     CALL SNDCMD          ;SEND COMMAND TO UDA
38 013750 012700 000264     MOV #3.*60.,R0          ;SET TIMEOUT FOR 3 MINUTES
39 013754 010501              MOV R5,R1
40 013756 062701 000040     ADD #C.TO,R1          ;PUT TIME IN CONTROLLER TABLE
41 013762 004737 022660     CALL SETTO
42 013766 000137 013366     JMP RSPNXT          ;NOW WAIT FOR END PACKET
43
44          ;RESPONSE REQUEST DISPATCH TABLE
45
46 013772 014030      RSPDSP: .WORD T1MSIZ          ; 0. SET UP FREE MEMORY FOR ADDRESS TESTING
47 013774 014150      .WORD T2DLL          ; 1. PROVIDE DIAGNOSTIC PROGRAM FOR DISK DRIVE
48 013776 014314      .WORD T2CMD          ; 2. GET MANUAL INTERVENTION COMMAND
49 014000 015150      .WORD T4MPRM          ; 3. TELL DATA PATTERN 16.
50 014002 015172      .WORD T4UPRM          ; 4. TELL UNIT PARAMETERS, CLEAR CONTENTS
51 014004 015520      .WORD T4BB1          ; 5. TELL BAD BLOCKS (FIRST 14)
52 014006 015550      .WORD T4BB2          ; 6. TELL BAD BLOCKS (LAST TWO)
53 014010 015600      .WORD T4SOFT          ; 7. ADD TO SOFT ERROR AND ECC COUNTS
    
```


54 014012 015626
55 014014 015646
56 014016 016010
57 014020 016076
58 014022 016344
59 014024 016466
60 014026 016576
61
62 000017

.WORD T4SEEK
.WORD T4MXFR
.WORD UTOTST
.WORD ERRMES
.WORD ERRMC
.WORD MESSAG
.WORD DONE
DSPSIZ=<.-RSPDSP>/2

: 8. ADD 1000 TO SEEK COUNT
: 9. ADD TO MEGABITS READ AND WRITE COUNTS
:10. TELL WHICH DRIVES TO TEST
:11. REPORT ERROR MESSAGE
:12. REPORT ERROR MESSAGE AND COUNT HARD ERROR
:13. PRINT A DESCRIPTIVE MESSAGE
:14. MARK DM PROGRAM AS NO LONGER RUNNING
;LEGAL NUMBERS ARE LOWER THAN THIS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

:NORMAL MAINTENANCE READ BUFFER DESCRIPTION

:BYTE OFFSET FROM
:START OF BUFFER

0
2
4
6
8
10
12
14
16
18
20
22

REQUEST NUMBER
DATA ARGUMENT #1
DATA ARGUMENT #2
DATA ARGUMENT #3
DATA ARGUMENT #4
DATA ARGUMENT #5
DATA ARGUMENT #6
DATA ARGUMENT #7
DATA ARGUMENT #8
DATA ARGUMENT #9
DATA ARGUMENT #10
DATA ARGUMENT #11

USED TO SELECT ROUTINE
R4 CONTAINS THIS ADDRESS

1			
2		;NORMAL PSEUDO-TERMINAL IN PACKET DESCRIPTION GIVEN IN RESPONSE TO ABOVE PACKET	
3			
4		;BYTE OFFSET FROM	
5		;START OF PACKET	
6		0	+-----+ : REQUEST NUMBER : ECHOED FROM REQUEST PACKET
7		2	+-----+ : DATA ARGUMENT #1 : R3 CONTAINS THIS ADDRESS
8		4	+-----+ : DATA ARGUMENT #2 : ALL DATA ARGUMENTS ARE RETURNED
9		6	+-----+ : DATA ARGUMENT #3 : CONTAINING ZEROS UNLESS
10		8	+-----+ : DATA ARGUMENT #4 : SPECIFICALLY INDICATED BY
11		10	+-----+ : DATA ARGUMENT #5 : RESPONSE ROUTINE.
12		12	+-----+ : DATA ARGUMENT #6 : : DATA ARGUMENT #7 :
13		14	+-----+ : DATA ARGUMENT #8 :
14		16	+-----+ : DATA ARGUMENT #9 :
15		18	+-----+ : DATA ARGUMENT #10 :
16		20	+-----+ : DATA ARGUMENT #11 :
17		22	+-----+ : DATA ARGUMENT #11 :
18			+-----+
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			

```

1      ;T1MSIZ - DM REQUEST 0
2
3      ;SET UP MEMORY FOR ADDRESS TESTING FROM UDA.
4      ;PLACE ADDRESS OF EACH LOCATION INTO EACH LOCATION IN FREE
5      ;MEMORY. RETURN FIRST LOCATION OF FREE MEMORY IN CMD.02 (LOW BITS)
6      ;AND CMD.03 (HIGH BITS). RETURN LAST LOCATION OF FREE MEMORY IN
7      ;CMD.04 AND CMD.05. ALSO RETURN FIRST EXISTANT LOCATION IN CMD.06
8      ;AND CMD.07; LAST EXISTANT LOCATION IN CMD.08 AND CMD.09.
9
10     ;INPUTS:
11     ;R5 - CONTROLLER TABLE ADDRESS
12     ;R4 - MESSAGE PACKET DATA ADDRESS (POINTING TO MSG.02)
13     ;R3 - COMMAND PACKET DATA ADDRESS (POINTING TO CMD.02)
14     ;OUTPUTS:
15     ;COMMAND PACKET CONTAINING:
16     ;1.(R3) LOW ADDRESS BITS OF FIRST WRITABLE ADDRESS
17     ;2.(R3) HIGH ADDRESS BITS OF FIRST WRITABLE ADDRESS
18     ;4.(R3) LOW ADDRESS BITS OF LAST WRITABLE ADDRESS
19     ;6.(R3) HIGH ADDRESS BITS OF LAST WRITABLE ADDRESS
20     ;8.(R3) LOW ADDRESS BITS OF FIRST READABLE ADDRESS
21     ;10.(R3) HIGH ADDRESS BITS OF FIRST READABLE ADDRESS
22     ;12.(R3) LOW ADDRESS BITS OF LAST READABLE ADDRESS
23     ;14.(R3) HIGH ADDRESS BITS OF LAST READABLE ADDRESS
24     ;Z SET
25
26 014030 013701 002174 T1MSIZ: MOV FFREE,R1 ;GET FIRST ADDRESS OF FREE MEMORY
27 014034 013702 002176      MOV FSIZE,R2 ;GET SIZE
28
29 ;FILL MEMORY WITH ADDRESS PATTERN
30
31 014040 010111 MEMFIL: MOV R1,(R1) ;WRITE DATA INTO LOCATION
32 014042 062701 000002      ADD #2,R1 ;INCREASE ADDRESS TO NEXT LOCATION
33 014046 005302      DEC R2 ;COUNT THE WORDS
34 014050 001373      BNE MEMFIL ;FILL ALL WORDS
35
36 ;SEND LOCATION OF FREE MEMORY TO UDA
37
38 014052 013723 002174      MOV FFREE,(R3)+ ;LOAD FIRST ADDRESS OF FREE MEMORY
39 014056 005023      CLR (R3)+ ;HIGH ORDER BITS ARE ZERO
40 014060 013700 002176      MOV FSIZE,R0 ;GET SIZE OF FREE MEMORY
41 014064 006300      ASL R0 ;CONVERT TO BYTES
42 014066 063700 002174      ADD FFREE,R0 ;COMPUTE LAST LOCATION
43 014072 162700 000002      SUB #2,R0
44 014076 010023      MOV R0,(R3)+ ;LOAD LAST LOCATION
45 014100 005023      CLR (R3)+ ;CLEAR HIGH ORDER BITS
46
47 ;SEND LOCATION OF READABLE MEMORY
48
49 014102 005023      CLR (R3)+ ;SEND ZERO AS START OF READABLE MEMORY
50 014104 005023      CLR (R3)+
51 014106 013700 002120      MOV L$HIMEM,R0 ;GET HIGH MEMORY ADDRESS
52 014112 005001      CLR R1 ;CLEAR HIGH BITS
53 014114 006300      ASL R0 ;SHIFT LEFT 6 PLACES
54 014116 006300      ASL R0
55 014120 006300      ASL R0
56 014122 006300      ASL R0
57 014124 006300      ASL R0
    
```

58 014126 006101
59 014130 006300
60 014132 006101
61 014134 052700 000076
62 014140 010023
63 014142 010123
64 014144 000264
65 014146 000207

ROL R1
ASL R0
ROL R1
BIS #76,R0
MOV R0,(R3)+
MOV R1,(R3)+
SEZ
RETURN

;SET LOW ORDER BITS
;PUT INTO BUFFER

```

1      ;T2DLL - DM REQUEST 1 . . . .
2
3      ;PROVIDE DIAGNOSTIC TO DOWNLINE LOAD INTO DISK DRIVE.
4
5      ;THE UDA MAY BE USED TO GET THE DIAGNOSTIC IF THE SYSTEM LOAD DEVICE
6      ;IS ON THE UDA. THIS ACTION WILL CAUSE A REINITIALIZATION OF THE UDA
7      ;AND THE RING STRUCTURE MOVED. SINCE THIS PROGRAM HAS NO WAY TO
8      ;DETERMINE IF THE UDA IS USED, IT WILL ALWAYS ASSUME IT IS USED AND
9      ;WILL INITIALIZE AND RELOAD THE DM PROGRAM AFTER THEREADING THE
10     ;DIAGNOSTIC. THE OUTPUTS OF THIS ROUTINE ARE STORED AND SENT TO THE
11     ;DM PROGRAM IN THE UTOTST REQUEST.
12
13     ;INPUTS:
14     R5 - CONTROLLER TABLE ADDRESS
15     R4 - MESSAGE DATA ADDRESS
16         (R4) DRIVE NUMBER
17     2.(R4) A VALUE THE DM PROGRAM WISHES RETURNED
18     4.(R4) REGION TO WHICH PROGRAM IS TO BE LOADED IN DISK
19     6.(R4) 2 WORD PROGRAM NAME IN RAD50
20     R3 - COMMAND DATA ADDRESS
21     ;OUTPUTS:
22     COMMAND PACKET COULD CONTAIN THE FOLLOWING:
23     (R3) ONE IF PROGRAM PROVIDED, TWO IF PROGRAM NOT AVAILABLE
24     2.(R3) DRIVE NUMBER
25     4.(R3) COPY OF THE VALUE FROM DM PROGRAM
26     6.(R3) REGION TO WHICH PROGRAM IS TO BE LOADED
27     8.(R3) ADDRESS OF FIRST BYTE TO BE DOWNLINE LOADED
28     10.(R3) HIGH ORDER BITS OF ADDRESS
29     12.(R3) BYTE COUNT OF PROGRAM TO BE DOWNLINE LOADED
30     Z SET
31     ;THIS PROGRAM WILL NOT SEND A COMMAND PACKET IN RESPONSE TO THIS REQUEST.
32     ;THE UDA WILL BE REINITIALIZED AND THE DM PROGRAM RELOADED. THEN THIS DATA
33     ;WILL BE APPENDED TO THE NEXT UTOTST REQUEST.
34
35     ;COPY REQUEST DATA TO STORAGE
36
37     T2DLL: CLR DLL ;CLEAR CONTROL WORD
38           MOV (R4)+,DLLDR ;DRIVE NUMBER
39           MOV (R4)+,DLLV ;VALUE FROM DM
40           MOV (R4)+,DLLR ;REGION
41           MOV (R4)+,DLLNAM ;PROGRAM NAME
42           MOV (R4)+,DLLNAM+2 ; (TWO WORDS)
43
44     ;RESET UDA AND READ DM PROGRAM
45
46     CLR @ (R5) ;RESET THE UDA
47     MOV FFREE,DLLADR ;GET ADDRESS WHERE PROGRAM
48     CLR DLLADR+2 ; TO BE STORED
49     MOV FSIZE,DLLSIZ ;SAVE CURRENT SIZE OF MEMORY
50     CALL RDDLL ;READ DLL PROGRAM FROM DATA FILE
51     BCC 1$ ;PROGRAM NOT FOUND IF CARRY SET
52     INC DLL ;RETURN 1 IF PROGRAM FOUND
53     1$: INC DLL ;RETURN 2 IF PROGRAM NOT FOUND
54     MOV DLLSIZ,FSIZE ;COMPUTE SIZE OF DLL PROGRAM
55     MOV FFREE,DLLSIZ ; AND RESTORE ORIGINAL FFREE
56     SUB DLLADR,DLLSIZ ; AND FSIZE VALUES
57     MOV DLLADR,FFREE
    
```

```

37 014150 005037 002412
38 014154 012437 002414
39 014160 012437 002416
40 014164 012437 002420
41 014170 012437 002430
42 014174 012437 002432
43
44
45
46 014200 005075 000000
47 014204 013737 002174 002422
48 014212 005037 002424
49 014216 013737 002176 002426
50 014224 004737 023674
51 014230 103002
52 014232 005237 002412
53 014236 005237 002412
54 014242 013737 002426 002176
55 014250 013737 002174 002426
56 014256 163737 002422 002426
57 014264 013737 002422 002174
    
```

58 014272 005726
59 014274 012701 000001
60 014300 004737 013134
61 014304 001402
62 014306 000137 013224
63 014312 000207

2\$:

TST (SP)+
MOV #1,R1
CALL RUNDM
BEQ 2\$
JMP RESPDM
RETURN

;POP RETURN ADDRESS OFF STACK
;RUN THE DM PROGRAM AGAIN

```

1      :T2CMD - DM REQUEST 2
2
3      :GET MANUAL INTERVENTION COMMAND
4
5      :INPUTS:
6          R5 - CONTROLLER TABLE ADDRESS
7          R4 - MESSAGE DATA ADDRESS
8              (R4) DRIVE NUMBER
9              2.(R4) OPERATION CODE
10             0 ON FIRST REQUEST FOR DRIVE. ECHO OF PREVIOUS RESPONSE ALL OTHER TIMES.
11             IF OPERATION CODE = 2
12             4.(R4) DATA BYTE READ (TO BE PRINTED)
13          R3 - COMMAND DATA ADDRESS
14      :OUTPUTS:
15          COMMAND DATA FILLED WITH THE FOLLOWING:
16              (R3) OPERATION CODE
17                  0 - EXIT
18                  1 - WRITE
19                  2 - READ
20                  3 - DIAGNOSE
21             IF OPERATION CODE = 1, 2 OR 3
22             2.(R3) REGION NUMBER
23             4.(R3) OFFSET INTO REGION
24             IF OPERATION CODE = 1
25             6.(R3) DATA BYTE
26          Z SET IF DATA RETURNED
27          Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
28
29 014314 032737 000200 002160 T2CMD: BIT #SM.MAN,SFPTBL+SO.BIT      ;LOOK AT MANUAL INTERVENTION MODE
30 014322 001002                BNE T2CMDM                      ;EXIT IF NOT WANTED
31 014324 000137 015122        JMP T2CMDX
32 014330                T2CMDM: MANUAL                          ;MANUAL INTERVENTION ALLOWED?
33 014332 104450                BCOMPLETE T2CMD0                  ;PRINT WARNING IF NOT
34 014334 103412                T2CMD9: PRINTF #T2WARN
35 014334 012746 010312                MOV #T2WARN,-(SP)
36 014340 012746 000001                MOV #1,-(SP)
37 014344 010600                MOV SP,R0
38 014346 104417                TRAP C$PNTF
39 014350 062706 000004                ADD #4,SP
40 014354 000137 015122                JMP T2CMDX
41 014360 012401                T2CMD0: MOV (R4)+,R1      ;GET DRIVE NUMBER
42 014362 012402                MOV (R4)+,R2      ;GET OPERATION CODE
43 014364 001056                BNE T2CMD2      ;BRANCH IF NOT ZERO
44 014366 004737 017414                CALL GTDRVT     ;GET DRIVE TABLE ADDRESS
45 014372 001401                BEQ 1$         ;CHECK IF DRIVE FOUND
46 014374 000207                RETURN        ;RETURN WITH Z CLEAR IF NOT
47 014376                1$: PRINTF #T2CMS1,D.UNIT(R4),(R5),(R4) ;PRINT DESCRIPTION
48 014376 011446                MOV (R4),-(SP)
49 014400 011546                MOV (R5),-(SP)
50 014402 016446 000002                MOV D.UNIT(R4),-(SP)
51 014406 012746 010414                MOV #T2CMS1,-(SP)
52 014412 012746 000004                MOV #4,-(SP)
53 014416 010600                MOV SP,R0
54 014420 104417                TRAP C$PNTF
55 014422 062706 000012                ADD #12,SP
    
```



```

43 014426          PRINTF #T2CMS2
    014426 012746 010523          MOV #T2CMS2,-(SP)
    014432 012746 000001          MOV #1,-(SP)
    014436 010600          MOV SP,R0
    014440 104417          TRAP C$PNTF
    014442 062706 000004          ADD #4,SP

44 014446          PRINTF #T2CMS3
    014446 012746 010640          MOV #T2CMS3,-(SP)
    014452 012746 000001          MOV #1,-(SP)
    014456 010600          MOV SP,R0
    014460 104417          TRAP C$PNTF
    014462 062706 000004          ADD #4,SP

45 014466          PRINTF #T2CMS4
    014466 012746 010707          MOV #T2CMS4,-(SP)
    014472 012746 000001          MOV #1,-(SP)
    014476 010600          MOV SP,R0
    014500 104417          TRAP C$PNTF
    014502 062706 000004          ADD #4,SP

46 014506 005037 002376          CLR T2WRR          ;CLEAR ALL STORAGE WORDS
47 014512 005037 002400          CLR T2WRO
48 014516 005037 002402          CLR T2DR

49
50 014522 022702 000002          T2CMD2: CMP #2,R2          ;SEE IF LAST OPERATION WAS READ
51 014526 001055          BNE T2CMDQ          ;BRANCH IF NOT TO QUESTION
52 014530 112737 000040 002172  MOVB #' ERRCHR          ;PRINT ONE SPACE
53 014536          PRINTF #ERRONE,#ERRCHR
    014536 012746 002172          MOV #ERRCHR,-(SP)
    014542 012746 005316          MOV #ERRONE,-(SP)
    014546 012746 000002          MOV #2,-(SP)
    014552 010600          MOV SP,R0
    014554 104417          TRAP C$PNTF
    014556 062706 000006          ADD #6,SP

54 014562 013701 002376          MOV T2WRR,R1          ;PRINT REGION
55 014566 004737 020420          CALL T2PNTW
56 014572 013701 002400          MOV T2WRO,R1          ;PRINT OFFSET
57 014576 004737 020420          CALL T2PNTW
58 014602 112737 000057 002172  MOVB #'/,ERRCHR          ;PRINT A SLASH
59 014610          PRINTF #ERRONE,#ERRCHR
    014610 012746 002172          MOV #ERRCHR,-(SP)
    014614 012746 005316          MOV #ERRONE,-(SP)
    014620 012746 000002          MOV #2,-(SP)
    014624 010600          MOV SP,R0
    014626 104417          TRAP C$PNTF
    014630 062706 000006          ADD #6,SP

60 014634 012401          MOV (R4)+,R1          ;PRINT THE DATA
61 014636 004737 020472          CALL T2PNTB
62 014642          PRINTF #ERRNL          ;END THE LINE
    014642 012746 005321          MOV #ERRNL,-(SP)
    014646 012746 000001          MOV #1,-(SP)
    014652 010600          MOV SP,R0
    014654 104417          TRAP C$PNTF
    014656 062706 000004          ADD #4,SP

63
64          ;NOW ASK FOR COMMAND INPUT
65
66 014662          T2CMDQ: GMANID T4OPT7,TEMP,A,-1,1,20,,NO          ;ASK FOR COMMAND
    014662 104443          TRAP C$GMAN
    
```

014664	000406						BR	10000\$
014666	002260						.WORD	TEMP
014670	000142						.WORD	T\$CODE
014672	004176						.WORD	T4OPT7
014674	177777						.WORD	-1
014676	000001						.WORD	T\$LOLIM
014700	000024						.WORD	T\$HILIM
014702								10000\$:
67	014702	012701	002260					:GET POINTER TO STRING
68	014706	112100						:GET COMMAND CHARACTER
69	014710	022700	000105					
70	014714	001415						
71	014716	022700	000104					
72	014722	001016						
73	014724	012713	000003					:STORE DIAGNOSE OPERATION CODE
74	014730	004737	020620					:GET REGION FROM COMMAND
75	014734	001402						
76	014736	010437	002402					
77	014742	013763	002402	000002	1\$:			
78	014750	004737	020620		T2CMDV:			:MAKE SURE AT END OF LINE
79	014754	001064						
80	014756	000461						
81								
82								:COMMAND MUST BE EITHER READ OR WRITE
83								
84	014760	012713	000002		T2CMD3:			:CHECK IF READ
85	014764	022700	000122					
86	014770	001415						
87	014772	022700	000127					:CHECK IF WRITE
88	014776	001053						: IF NOT - ERROR
89	015000	012713	000001					
90	015004	004737	020620					:GET DATA BYTE
91	015010	001446						:ERROR IF NO DATA
92	015012	162700	000002					
93	015016	003043						:OR GREATER THAN TWO DIGITS
94	015020	010463	000006					:STORE DATA BYTES IN BUFFER
95	015024	013763	002376	000002	T2CMDR:			:PUT REGION AND OFFSET
96	015032	013763	002400	000004				: INTO BUFFER
97	015040	021302						: IF SO,
98	015042	001002						
99	015044	005263	000004					: INCREMENT OFFSET
100	015050	004737	020620		T2CMDN:			
101	015054	001411						
102	015056	010463	000002					
103	015062	005063	000004					
104	015066	004737	020620					
105	015072	001402						
106	015074	010463	000004					
107	015100	004737	020620		T2CMDW:			
108	015104	001010						
109	015106	016337	000002	002376				:SAVE REGION
110	015114	016337	000004	002400				:SAVE OFFSET
111	015122	000264			T2CMDX:			
112	015124	000207						
113	015126				T2CMDE:			:REPORT ERROR MESSAGE
	015126	012746	011022				MOV	#T2CMS5,-(SP)
	015132	012746	000001				MOV	#1,-(SP)

015136 01060C
015140 104417
015142 062706 000004
114 015146 000645

BR T2CMDQ

;GO ASK AGAIN

MOV SP,R0
TRAP C\$PNTF
ADD #4,SP

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 015150 012701 000021
19 015154 012702 002310
20 015160 012223
21 015162 005301
22 015164 001375
23 015166 000264
24 015170 000207

```
:T4MPRM - DM REQUEST 3  
:REQUEST FOR TEST 4 CONTENTS OF DATA PATTERN 16.  
:INPUTS:  
:   R5 - CONTROLLER TABLE ADDRESS  
:   R4 - MESSAGE DATA ADDRESS  
:         (NO DATA)  
:   R3 - COMMAND DATA ADDRESS  
:OUTPUTS:  
:   COMMAND DATA FILLED WITH THE FOLLOWING:  
:   (R3) NUMBER OF WORDS IN DATA PATTERN 16  
:   2.(R3) DATA IN PATTERN 16  
:   |  
:   32.(R3) ..  
:   Z SET  
T4MPRM: MOV #17.,R1           ;GET COUNT  
        MOV #PAT16C,R2       ; AND ADDRESS OF PATTERN 16 PARAMETERS  
1$:    MOV (R2)+,(R3)+       ;COPY THE DATA TO BUFFER  
        DEC R1  
        BNE 1$  
        SEZ  
        RETURN               ;RETURN WITH Z SET
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```
:T4UPRM - DM REQUEST 4  
:REQUEST FOR TEST 4 UNIT PARAMETERS AND CLEAR STATISTICS FOR DRIVE  
:INPUTS:  
R5 - CONTROLLER TABLE ADDRESS  
R4 - MESSAGE DATA ADDRESS  
  (R4) DRIVE NUMBER  
  2.(R4) DRIVE SERIAL NUMBER  
  :  
  6.(R4)  
  8.(R4) HDA SERIAL NUMBER  
  :  
 14.(R4)  
R3 - COMMAND DATA ADDRESS  
:OUTPUTS:  
COMMAND DATA FILLED WITH THE FOLLOWING:  
(R3) PARAMETER BITS (1 FOR TRUE)  
  BIT 14 - INITIAL WRITE  
  BIT 13 - DIAGNOSTIC CYLINDERS  
  BIT 12 - ECC CORRECTION  
  BIT 11 - READ ONLY  
  BIT 10 - WRITE ONLY  
  BIT 9 - RETRIES  
  BIT 8 - TRACK/GROUP AND CYLINDERS SPECIFIED  
  BIT 7 - (NOT USED)  
  BIT 6 - SEQUENTIAL SEEKS  
  BIT 5 - BEGIN/END SETS SPECIFIED  
  BIT 4 - TRACK SPECIFIED (0 - GROUPS SPECIFIED)  
           HAS MEANING ONLY WHEN BIT 5 IS ZERO  
  BIT 3 - WRITE CHECKS ENABLED  
  BIT 2 - WRITE CHECKS ALWAYS  
  BIT 1 - DATA COMPARES ENABLED  
  BIT 0 - DATA COMPARE ALWAYS  
  2.(R3) DATA PATTERN NUMBER  
IF PARAMETER BIT 5 SET  
  4.(R3) COUNT OF BEGIN/END SETS  
  6.(R3) END BLOCK (2 WORDS) THEN BEGIN BLOCK (2 WORDS)  
           1 TO 4 SETS  
           OR  
           IF COUNT OF BEGIN/END BLOCKS = 0  
36.(R3) START CYLINDER (2 WORDS) THEN END CYLINDER (2 WORDS)  
           END CYLINDER A NEGATIVE VALUE IF TO TEST ENTIRE AREA  
IF PARAMETER BIT 5 CLEAR  
  4.(R3) STARTING CYLINDER  
  6.(R3) (2 WORDS)  
  8.(R3) ENDING CYLINDER (2 WORDS)  
 10.(R3) NEGATIVE FOR ALL CYLINDERS  
 12.(R3) NUMBER OF TRACKS OR GROUPS SPECIFIED  
 14.(R3) 1 TO 7 TRACK OR GROUP NUMBERS  
           DETERMINED BY PARAMETER BIT 4  
26.(R3)  
Z SET IF DATA RETURNED  
Z CLEAR IF UNIT NUMBER NOT ON THIS CONTROLLER  
D.XFRW, D.XFRR, D.HERR, D.SERR, D.SEEK AND D.ECC CLEARED  
IN DRIVE TABLE
```

1	015172	012401			T4UPRM: MOV (R4)+,R1	:GET DRIVE NUMBER
2	015174	010402			MOV R4,R2	:SAVE DATA ADDRESS
3	015176	004737	017414		CALL GTDRVT	:GET DRIVE TABLE ADDRESS
4	015202	001145			BNE T4UPRE	:CHECK IF DRIVE FOUND
5	015204	012264	000200		MOV (R2)+,D.SERN(R4)	:COPY DRIVE SERIAL NUMBER TO DRIVE TABLE
6	015210	012264	000202		MOV (R2)+,D.SERN+2(R4)	
7	015214	012264	000204		MOV (R2)+,D.SERN+4(R4)	
8	015220	016401	000004		MOV D.PRM(R4),R1	:GET PARAMETER BITS
9	015224	042701	140200		BIC #D.ZERO,R1	:CLEAR SOME BITS
10	015230	032737	000020	002224	BIT #ISTRTH,IFLAGS	:IF FIRST TIME TEST 4 BEING RUN
11	015236	001406			BEQ 2\$: AFTER A START COMMAND
12	015240	032737	040000	002160	BIT #SM.IW,SFPTBL+SO.BIT	:GET INITIAL WRITE BIT
13	015246	001402			BEQ 2\$	
14	015250	052701	040000		BIS #D.IW,R1	:MOVE INTO PARAMETER BITS
15	015254	010123			2\$: MOV R1,(R3)+	:PUT INTO BUFFER
16	015256	016423	000006		MOV D.PAT(R4),(R3)+	:PUT PATTERN NUMBER IN BUFFER
17	015262	032701	000040		BIT #D.BE,R1	:CHECK BEGIN/END PARAMETER BIT
18	015266	001421			BEQ 10\$:BRANCH IF NOT SET
19						
20					:RETURN BEGIN/END SETS	
21						
22	015270	012701	000004		MOV #4,R1	:GET COUNT OF SETS
23	015274	010402			MOV R4,R2	:GET INDEX INTO DRIVE TABLE
24	015276	062702	000112		ADD #D.BEC,R2	
25	015302	012223			MOV (R2)+,(R3)+	:COPY COUNT
26	015304	016223	000004		1\$: MOV 4(R2),(R3)+	
27	015310	016223	000006		MOV 6(R2),(R3)+	
28	015314	012223			MOV (R2)+,(R3)+	
29	015316	012223			MOV (R2)+,(R3)+	
30	015320	062702	000004		ADD #4,R2	
31	015324	005301			DEC R1	
32	015326	001366			BNE 1\$	
33	015330	000456			BR T4UPRX	
34						
35	015332	032764	000400	000004	10\$: BIT #D.CYL,D.PRM(R4)	:LOOK AT D.CYL BIT
36	015334	001441			BEQ 20\$:BRANCH IF NOT SET
37						
38					:RETURN TRACKS/GROUPS AND CYLINDERS	
39						
40	015342	005764	000112		TST D.BEC(R4)	:CHECK IF ANY TRACKS/GROUPS
41	015346	001421			BEQ 25\$:BRANCH IF NONE
42	015350	012701	000004		MOV #4,R1	:COUNT OF CYLINDER WORDS
43	015354	010402			MOV R4,R2	
44	015356	062702	000154		ADD #D.BCYL,R2	
45	015362	012223			11\$: MOV (R2)+,(R3)+	:CYLINDERS
46	015364	005301			DEC R1	
47	015366	001375			BNE 11\$	
48	015370	012701	000010		MOV #8,R1	
49	015374	010402			MOV R4,R2	
50	015376	062702	000112		ADD #D.BEC,R2	
51	015402	012223			12\$: MOV (R2)+,(R3)+	:TRACKS/GROUPS
52	015404	005301			DEC R1	
53	015406	001375			BNE 12\$	
54	015410	000426			BR T4UPRX	
55						
56					:RETURN CYLINDERS ONLY	
57						

```
58 015412 052763 000040 177774 25$: BIS #D.BE,-4(R3)           ;SET D.BE FOR DM PROGRAM
59 015420 005023                CLR (R3)+           ;SEND ZERO BEGIN/END COUNT
60 015422 012701 000004        MOV #4,R1
61 015426 010402                MOV R4,R2
62 015430 062702 000154        ADD #D.BCYL,R2
63 015434 012223                26$: MOV (R2)+,(R3)+       ;CYLINDERS
64 015436 005301                DEC R1
65 015440 001375                BNE 26$
66 015442 000411                BR T4UPRX
67
68                                ;RETURN ENTIRE AREA
69
70 015444 052763 000040 177774 20$: BIS #D.BE,-4(R3)           ;SET D.BE FOR DM PROGRAM
71 015452 005023                CLR (R3)+           ;BEGIN/END COUNT OF ZERO
72 015454 005023                CLR (R3)+           ;START CYLINDER OF ZERO
73 015456 005023                CLR (R3)+
74 015460 005023                CLR (R3)+           ;END CYLINDER NEGATIVE
75 015462 012723 177777        MOV #-1,(R3)+
76 015466 005064 000164        T4UPRX: CLR D.XFRW(R4)       ;CLEAR STATISTICS
77 015472 005064 000166        CLR D.XFRR(R4)
78 015476 005064 000170        CLR D.HERR(R4)
79 015502 005064 000172        CLR D.SERR(R4)
80 015506 005064 000174        CLR D.SEEK(R4)
81 015512 005064 000176        CLR D.ECCC(R4)
82 015516 000207        T4UPRE: RETURN
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 015520 011401
23 015522 004737 017414
24 015526 001007
25 015530 062704 000010
26 015534 012701 000035
27 015540 012423
28 015542 005301
29 015544 001375
30 015546 000207

```

:T4BB1 - DM REQUEST 5
:REQUEST FOR FIRST 14 BAD BLOCKS
:INPUTS:
R5 - CONTROLLER TABLE ADDRESS
R4 - MESSAGE DATA ADDRESS
(R4) DRIVE NUMBER
R3 - COMMAND DATA ADDRESS
:OUTPUTS:
COMMAND DATA FILLED WITH BAD BLOCKS
(R3) COUNT OF BAD BLOCKS
2.(R3) BAD BLOCK 1 (LOW)
4.(R3) (HIGH)
:
56.(R3) BAD BLOCK 14 (LOW)
58.(R3) (HIGH)
Z SET IF DATA RETURNED
Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
:
T4BB1: MOV (R4),R1 :GET DRIVE NUMBER
CALL GTDRVT :GET DRIVE TABLE ADDRESS
BNE T4BB1E :CHECK IF DRIVE FOUND
ADD #D.BB,R4 :INCREASE ADDRESS TO DATA TO COPY
MOV #<1+<14.*2>>,R1 :GET COUNT OF WORDS
1$: MOV (R4)+,(R3)+ :COPY THE WORDS
DEC R1
BNE 1$
T4BB1E: RETURN
    
```



```
1      ;T4BB2 - DM REQUEST 6
2      ;
3      ;REQUEST LAST TWO BAD BLOCKS
4      ;
5      ;INPUTS:
6      ;   R5 - CONTROLLER TABLE ADDRESS
7      ;   R4 - MESSAGE DATA ADDRESS
8      ;         (R4) DRIVE NUMBER
9      ;   R3 - COMMAND DATA ADDRESS
10     ;OUTPUTS:
11     ;   COMMAND DATA FILLED WITH BAD BLOCKS 15 AND 16
12     ;   Z SET IF DATA RETURNED
13     ;   Z CLEAR IF UNIT NUMBER NOT ON THIS CONTROLLER
14     ;
15 015550 011401      T4BB2:  MOV (R4),R1           ;GET DRIVE NUMBER
16 015552 004737 017414  CALL GTDRVT         ;GET DRIVE TABLE ADDRESS
17 015556 001007      BNE T4BB2E        ;CHECK IF DRIVE FOUND
18 015560 062704 000102  ADD #D.BB15,R4     ;INCREASE ADDRESS TO DATA TO COPY
19 015564 012701 000004  MOV #4,R1          ;GET COUNT OF WORDS
20 015570 012423      1$:  MOV (R4)+,(R3)+    ;COPY THE WORDS
21 015572 005301      DEC R1
22 015574 001375      BNE 1$
23 015576 000207      T4BB2E: RETURN
```

1
2
3
4
5
6
7
8
9
10
11
12
13 015600 012401
14 015602 010402
15 015604 004737 017414
16 015610 001005
17 015612 062264 000172
18 015616 062264 000176
19 015622 000264
20 015624 000207

```
:T4SOFT - DM REQUEST 7  
:  
:ADD TO SOFT ERROR AND ECC COUNTS  
:  
:INPUTS:  
: R5 - CONTROLLER TABLE ADDRESS  
: R4 - MESSAGE DATA ADDRESS  
: (R4) DRIVE NUMBER  
: 2.(R4) VALUE TO ADD TO SOFT ERROR COUNT  
: 4.(R4) VALUE TO ADD TO ECC COUNT  
:  
: R3 - COMMAND DATA ADDRESS  
:  
T4SOFT: MOV (R4)+,R1 ;GET DRIVE NUMBER  
MOV R4,R2 ;SAVE DATA ADDRESS  
CALL GTDRVT ;GET DRIVE TABLE ADDRESS  
BNE 1$ ;CHECK IF DRIVE FOUND  
ADD (R2)+,D.SERR(R4) ;ADD TO SOFT ERROR COUNT  
ADD (R2)+,D.ECCC(R4) ;ADD TO ECC COUNT  
SEZ  
1$: RETURN
```

1 015626
2
3
4
5
6
7
8
9
10
11
12 015626 011401
13 015630 004737 017414
14 015634 001003
15 015636 005264 000174
16 015642 000264
17 015644 000207

T4SEEK:
: DM REQUEST 8.
: RECORD 1000 SEEKS COMPLETED ON DRIVE
: INPUTS:
: R5 - CONTROLLER TABLE ADDRESS
: R4 - MESSAGE DATA ADDRESS
: (R4) DRIVE NUMBER
: R3 - COMMAND DATA ADDRESS
: MOV (R4),R1 ; GET DRIVE NUMBER
: CALL GTDRVT ; GET DRIVE TABLE ADDRESS
: BNE SEKERE ; CHECK IF DRIVE FOUND
: INC D.SEEK(R4) ; COUNT THE BITS TRANSFERRED
: SEZ ; NORMAL RETURN
SEKERE: RETURN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

```

:T4MXFR - DM REQUEST 9.
:RECORD 1M BITS TRANSFERRED ON UNIT. COMPARE TO TRANSFER LIMIT AND
:REPORT LIMIT REACHED.
:INPUTS:
R5 - CONTROLLER TABLE ADDRESS
R4 - MESSAGE DATA ADDRESS
(R4) DRIVE NUMBER
2.(R4) VALUE TO ADD TO READ COUNT
4.(R4) VALUE TO ADD TO WRITE COUNT
R3 - COMMAND DATA ADDRESS
:OUTPUTS:
(R3) BIT 15 SET IF TRANSFER LIMIT REACHED
MESSAGE PRINTED IF TRANSFER LIMIT REACHED
Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
    
```

```

18 015646 010402          T4MXFR: MOV R4,R2          ;GET MESSAGE DATA ADDRESS
19 015650 011401          MOV (R4),R1        ;GET DRIVE NUMBER
20 015652 004737 017414   CALL GTDRVT       ;GET DRIVE TABLE ADDRESS
21 015656 001053          BNE MXFERE       ;CHECK IF DRIVE FOUND
22 015660 066264 000002 000166 ADD 2(R2),D.XFRR(R4) ;ADD MEGABITS READ
23 015666 066264 000004 000164 ADD 4(R2),D.XFRW(R4) ;ADD MEGABITS WRITTEN
24 015674 005737 002156   TST SFPTBL+SO.XL ;SEE IF LIMIT SPECIFIED
25 015700 001441          BEQ MXFERX       ;BRANCH IF NOT
26 015702 026437 000166 002156 CMP D.XFRR(R4),SFPTBL+SO.XL ;CHECK IF LIMIT REACHED
27 015710 103435          BLO MXFERX       ;BRANCH IF LIMIT NOT REACHED
28 015712          RFLAGS R0 ;CHECK FLAGS
29 015714 032700 000040          BIT #IDU,R0 ;SEE IF DROPPING UNITS IS INHIBITED
30 015720 001031          BNE MXFERX
31 015722 052713 100000          BIS #BIT15,(R3) ;SET DROP UNIT BIT
32 015726 042765 000010 000014 BIC #CT.MSG,C.FLG(R5) ;CLEAR MESSAGE RECEIVED FLAG
33 015734          PRINTF #MESSG,D.UNIT(R4),(R5),(R4) ;PRINT TESTING DONE
34 015734 011446          MOV (R4),-(SP)
35 015736 011546          MOV (R5),-(SP)
36 015740 016446 000002          MOV D.UNIT(R4),-(SP)
37 015744 012746 010131          MOV #MESSG,-(SP)
38 015750 012746 000004          MOV #4,-(SP)
39 015754 010600          MOV SP,R0
40 015756 104417          TRAP C$PNTF
41 015760 062706 000012          ADD #12,SP
42 015764          PRINTF #MXFERP
43 015764 012746 006425          MOV #MXFERP,-(SP)
44 015770 012746 000001          MOV #1,-(SP)
45 015774 010600          MOV SP,R0
46 015776 104417          TRAP C$PNTF
47 016000 062706 000004          ADD #4,SP
48 016004 000264          MXFERX: SEZ ;NORMAL RETURN
49 016006 000207          MXFERE: RETURN
    
```

```

1      :UTOTST - DM REQUEST 10
2
3      :TELL DM PROGRAM WHICH DRIVES ARE SELECTED FOR TESTING
4
5      :INPUTS:
6          R5 - CONTROLLER TABLE ADDRESS
7          R4 - MESSAGE DATA ADDRESS
8              (NO DATA)
9          R3 - COMMAND DATA ADDRESS
10     :OUTPUTS:
11     COMMAND PACKET CONTAINING UP TO 8 DRIVE NUMBERS.
12     LIST IS ENDED BY A WORD WITH BIT 15 SET.
13     FOLLOWING LIST IS THE INFORMATION FROM T2DLL REQUEST IF APPLICABLE.
14     Z SET
15
16 016010 010504      UTOTST: MOV R5,R4          :GET ADDRESS OF CONTROLLER TABLE
17 016012 062704 000020 ADD #C.DR0,R4      :BUMP TO DRIVE TABLE POINTERS
18 016016 012702 000010 MOV #8.,R2         :GET COUNT OF PORTS
19 016022 012400      UTOT1: MOV (R4)+,R0      :SEE IF DRIVE TABLE POINTER EXISTS
20 016024 001406      BEQ UTOT2          :BRANCH IF NOT
21 016026 005760 000002 TST D.UNIT(R0)    :LOOK IF UNIT AVAILABLE FOR TESTING
22 016032 100773      BMI UTOT1
23 016034              ASSUME DT.AVL EQ BIT15
24 016034 011023      MOV (R0),(R3)+      :LOAD DRIVE NUMBER FROM TABLE
25 016036 005302      DEC R2              :COUNT THE PORTS
26 016040 001370      BNE UTOT1          :REPEAT FOR EACH PORT
27 016042 012723 100000 UTOT2: MOV #BIT15,(R3)+      :TERMINATE LIST
28 016046 013723 002412 MOV DLL,(R3)+      :GET DLL CONTROL WORD
29 016052 001407      BEQ UTOT4          : IF NON-ZERO
30 016054 012701 002414 MOV #DLLDR,R1      : TRANSFER ALL DLL WORDS INTO BUFFER
31 016060 012702 000020 MOV #<DLLNAM+4-DLLDR>,R2
32 016064 012123      UTOT3: MOV (R1)+,(R3)+
33 016066 005302      DEC R2
34 016070 001375      BNE UTOT3
35 016072 000264      UTOT4: SEZ
36 016074 000207      RETURN          :RETURN WITH Z SET
    
```

```

1      :ERRMES - DM REQUEST 11
2
3      :PRINT AN ERROR MESSAGE
4
5      :INPUTS:
6          R5 - CONTROLLER TABLE ADDRESS
7          R4 - MESSAGE DATA ADDRESS
8              (R4) ERROR PC IN DM PROGRAM
9              2.(R4) < 9:8 > ERROR TYPE
10             < 7:0 > ERROR NUMBER
11             4.(R4) DRIVE NUMBER (-1 IF NOT GIVEN)
12             6.(R4) MESSAGE POINTER
13             8.(R4) OPTIONAL PARAMETERS FOR ERROR PRINT ROUTINE
14             10.(R4)
15             ..
16             ..
17             58.(R4) ..
18          R3 - COMMAND DATA ADDRESS
19      :OUTPUTS:
20          COMMAND PACKET CONTAINING THE FOLLOWING:
21              (R3) - BIT 15 SET IF FATAL ERROR TO INDICATE DRIVE SHOULD NO LONGER BE TESTED
22              Z SET TO INDICATE DATA RETURNED
23              Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
24
25 016076 122764 000001 000003 ERRMES: CMPB #1,3(R4)          :CHECK IF FATAL ERROR
26 016104 100406          BMI 4$                          :BRANCH IF NOT
27 016106          RFLAGS R0                                :LOOK AT FLAGS
28 016110 032700 000040          BIT #IDU,R0                :SEE IF ALLOWED TO DROP UNITS
29 016114 001012          BNE 3$                          :BRANCH IF NOT
30 016116 052713 100000          BIS #BIT15,(R3)           :SET DROP DRIVE BIT
31 016122 122764 000003 000003 4$: CMPB #3,3(R4)          :SEE IF SOFT ERROR
32 016130 001004          BNE 3$                          :BRANCH IF NOT
33 016132 032737 000400 0C2160          BIT #SM.SSF,SO.BIT+SFPTBL :SEE IF SOFT ERRORS SUPPRESSED
34 016140 001077          BNE ERRMSX                       :DON'T PRINT IF SO
35 016142 042765 000010 000014 3$: BIC #CT.MSG,C.FLG(R5)   :CLEAR MESSAGE RECEIVED FLAG
36 016150 022737 000004 002226          CMP #4,TNUM                :IF TEST # 4,
37 016156 001004          BNE 1$
38 016160 032737 001000 002160          BIT #SM.LOG,SFPTBL+SO.BIT :SEE IF LOG BEING USED
39 016166 001003          BNE ERRMSL
40 016170 004737 020746          1$: CALL PNTERR                :IF NOT, PRINT THE ERROR MESSAGE
41 016174 000461          BR ERRMSX
42
43 016176 005737 002404          ERRMSL: TST LBUFS                :SEE IF LOG BUFFER ESTABLISHED
44 016202 001014          BNE 1$                          :LBUFS CONTAINS ADDRESS IF ESTABLISHED
45 016204 013701 002174          MOV FFREE,R1              :GET START ADDRESS OF BUFFER
46 016210 010137 002404          MOV R1,LBUFS                :INITIALIZE BUFFER STORAGE
47 016214 010137 002406          MOV R1,LBUFN               :SAVE ADDRESS WHERE TO ADD
48 016220 063701 002176          ADD FSIZE,R1              :COMPUTE END OF STORAGE AREA
49 016224 063701 002176          ADD FSIZE,R1
50 016230 010137 002410          MOV R1,LBUFE
51 016234 013701 002406          1$: MOV LBUFN,R1                :GET ADDRESS OF DATA STORAGE AREA
52 016240 062737 000074 002406          ADD #HC.BSZ,LBUFN         :ADD BYTES OF STORAGE NEEDED
53 016246 023737 002406 002410          CMP LBUFN,LBUFE         :SEE IF ENOUGH ROOM
54 016254 103017          BHS 3$                          :BRANCH IF NOT
55 016256 010521          MOV R5,(R1)+              :STORE CONTROLLER TABLE ADDRESS
56 016260 012700 000035          MOV #<HC.BSZ-2>/2,R0    :GET COUNT OF REST OF DATA IN WORDS
    
```

```
57 016264 012421          2$:  MOV (R4)+,(R1)+          ;STORE DATA
58 016266 005300          DEC R0
59 016270 001375          BNE 2$
60 016272          PRINTF #LOGDAT          ;TELL OPERATOR DATA IN BUFFER
    016272 012746 011514          MOV #LOGDAT,-(SP)
    016276 012746 000001          MOV #1,-(SP)
    016302 010600          MOV SP,R0
    016304 104417          TRAP C$PNTF
    016306 062706 000004          ADD #4,SP
61 016312 000412          BR ERRMSX
62 016314 010137 002406    3$:  MOV R1,LBUFN          ;RESTORE OLD VALUE OF LBUFN
63 016320          PRINTF #LOGFUL          ;TELL OPERATOR LOG IS FULL
    016320 012746 011562          MOV #LOGFUL,-(SP)
    016324 012746 000001          MOV #1,-(SP)
    016330 010600          MOV SP,R0
    016332 104417          TRAP C$PNTF
    016334 062706 000004          ADD #4,SP
64 016340 000264          ERRMSX: SEZ
65 016342 000207          RETURN
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

```

:ERRMC - DM REQUEST 12.
:REPORT AN ERROR MESSAGE IDENTICAL TO DM REQUEST ERRMES
:THEN ADD ONE TO THE ERROR COUNT FOR THE DRIVE AND SEE IF
:ERROR LIMIT REACHED.

:INPUTS:
R5 - CONTROLLER TABLE ADDRESS
R4 - MESSAGE DATA ADDRESS
(R4) ERROR PC IN DM PROGRAM
2.(R4) < 9:8 > ERROR TYPE
      < 7:0 > ERROR NUMBER
4.(R4) DRIVE NUMBER (-1 IF NOT GIVEN)
6.(R4) MESSAGE POINTER
8.(R4) OPTIONAL PARAMETERS FOR ERROR PRINT ROUTINE
10.(R4) ..
      ..
      ..
58.(R4) ..
R3 - COMMAND DATA ADDRESS
    
```

```

:OUTPUTS:
COMMAND PACKET CONTAINING THE FOLLOWING:
(R3) BIT 15 SET IF ERROR COUNT REACHED
      TO INDICATE DRIVE SHOULD NO LONGER BE TESTED.
Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
Z SET TO INDICATE DATA RETURNED
    
```

```

ERRMC: PUSH R4
      MOV R4,-(SP)
      CALL ERRMES           : CALL REQUEST ERRMES
      BNE 4$
      POP R4
      MOV (SP)+,R4
      MOV 4(R4),R1         : GET DRIVE NUMBER
      MOVB 3(R4),R2        : GET ERROR TYPE
      CALL GTDRVT         : GET DRIVE TABLE
      BNE 5$              : EXIT IF NO TABLE FOR UNIT
      CMP #2,R2           : CHECK IF HARD ERROR
      BNE 3$              : BRANCH IF NOT
      INC D.HERR(R4)      : COUNT THE ERROR
      CMP D.HERR(R4),SFPTBL+SO.EL : CHECK IF AT LIMIT
      BLO 3$              : IF LIMIT REACHED, BRANCH
      RFLAGS R0          : LOOK AT THE FLAGS
      BIT #IDU,R0        : SEE IF DROPPING UNITS INHIBITED
      BNE 3$              : BRANCH IF SO
      PRINTF #ERRLIM,R1  : PRINT LIMIT REACHED
      MOV R1,-(SP)
      MOV #ERRLIM,-(SP)
      MOV #2,-(SP)
      MOV SP,R0
      TRAP C$PNTF
      ADD #6,SP

      BIS #BIT15,(R3)    : SET STOP TESTING BIT
3$: SEZ                  : SET Z FOR NORMAL RETURN
      RETURN             : RETURN TO CALLING PROGRAM
    
```

002154

006502

100000

49 016460
• 016460 012604
50 016462 000244
51 016464 000207

4\$: POP R4
MOV (SP)+,R4
5\$: CLZ
RETURN

: ADJUST STACK
: FLAG AS ERROR
: RETURN TO CALLING PROGRAM

```

1      :MESSAG - DM REQUEST 13.
2
3      :PRINT A MESSAGE WITH HEADER AS FOLLOWS:
4      :   UNIT XX UDA AT XXXXXX DRIVE XXX
5
6      :INPUTS:
7      :   R5 - CONTROLLER TABLE ADDRESS
8      :   R4 - MESSAGE DATA ADDRESS
9      :       (R4) DRIVE NUMBER
10     :       2.(R4) MESSAGE POINTER
11     :       4.(R4) OPTIONAL MESSAGE PARAMETERS
12
13     :
14     :   58.(R4) COMMAND DATA ADDRESS
15
16 016466 042765 000010 000014 MESSAG: BIC #CT.MSG,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
17 016474 012401                MOV (R4)+,R1                ;GET DRIVE NUMBER
18 016476                PUSH R4                ;SAVE DATA POINTER
19 016476 010446                MOV R4,-(SP)
20 016500 004737 017414        CALL GTDRVT                ;GET DRIVE TABLE ADDRESS
21 016504 001032                BNE 1$                ;CHECK IF DRIVE FOUND
21 016506                PRINTX #MESSG,D.UNIT(R4),(R5),(R4) ;PRINT HEADER
22 016506 011446                MOV (R4),-(SP)
23 016510 011546                MOV (R5),-(SP)
24 016512 016446 000002        MOV D.UNIT(R4),-(SP)
25 016516 012746 010131        MOV #MESSG,-(SP)
26 016522 012746 000004        MOV #4,-(SP)
27 016526 010600                MOV SP,R0
28 016530 104415                TRAP C$PNTX
29 016532 062706 000012        ADD #12,SP
30 016536                POP R4
31 016536 012604                MOV (SP)+,R4
32 016540 012402                MOV (R4)+,R2                ;GET MESSAGE POINTER
33 016542 006302                ASL R2                ;DOUBLE TO MAKE BYTE OFFSET
34 016544 063702 002212        ADD DMPROG,R2            ;ADD TO START OF MESSAGE STRINGS
35 016550 067702 163436        ADD @DMPROG,R2          ;ADD SIZE OF MAIN PROGRAM
36 016554 105712                TSTB (R2)                ;CHECK FIRST BYTE
37 016556 001001                BNE 2$                ;IF ZERO
38 016560 005202                INC R2                ; INCREMENT TO NEXT BYTE
39 016562 004737 016602        2$: CALL OSTRNG                ;OUTPUT ACCORDING TO STRING
40 016566 000264                SEZ
41 016570 000207                RETURN
42
43 1$: POP R4
44 016572                MOV (SP)+,R4
45 016574 012604                RETURN
46 016574 000207
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14

```
:DONE - DM REQUEST 14  
:MARK DM PROGRAM AS NO LONGER RUNNING  
:INPUTS:  
:   R5 - CONTROLLER TABLE ADDRESS  
:   R4 - MESSAGE DATA ADDRESS  
:         (NO DATA)  
:   R3 - COMMAND DATA ADDRESS  
:OUTPUTS:  
:   Z CLEAR TO DROP UNIT FROM TESTING  
DONE:   CLZ           ;DROP UNIT FROM TESTING  
        RETURN
```

13 016576 000244
14 016600 000207

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

```

:OSTRNG
:OUTPUT A MESSAGE ACCORDING TO A FORMAT STRING
:INPUTS:
:   R2 - ADDRESS OF START OF FORMAT STRING
:   R4 - ADDRESS OF PARAMETERS
:OUTPUTS:
:   R2 AND R4 UPDATED TO END OF STRING AND PARAMETERS

OSTRNG: MOVB (R2)+,R1           ;GET CONTROL CHARACTER
        BEQ OSTRE             ;EXIT IF NULL CHARACTER
        MOV #ERRC,R0         ;GET POINTER TO CHARACTER TABLE
NCONS:  CMPB R1,(R0)         ;COMPARE CHARACTER WITH TABLE ENTRY
        BEQ NCONF           ;BRANCH IF MATCH FOUND
        TSTB (R0)+         ;INCREMENT POINTER
        BNE NCONS          ;CONTINUE SEARCH IF NOT END OF TABLE
        PRINTF #ERRME1     ;REPORT BAD CONTROL CHARACTER
                                MOV #ERRME1,-(SP)
                                MOV #1,-(SP)
                                MOV SP,R0
                                TRAP C$PNTF
                                ADD #4,SP

NCONF:  BR OSTRE
        SUB #ERRC,R0       ;GET INCREMENT INTO TABLE
        ASL R0             ;DOUBLE TO WORD COUNT
        CALL @ERRD(R0)    ;DISPATCH TO PRINT ROUTINE
OSTRE:  BR OSTRNG        ;GET NEXT
        RETURN
    
```

```

11 016602 112201
12 016604 001425
13 016606 012700 017216
14 016612 120110
15 016614 001413
16 016616 105720
17 016620 001374
18 016622
   016622 012746 006332
   016626 012746 000001
   016632 010600
   016634 104417
   016636 062706 000004
19 016642 000406
20 016644 162700 017216
21 016650 006300
22 016652 004770 017230
23 016656 000751
24 016660 000207
    
```

```

1          ;CONTROL CHARACTER WAS A QUOTE. PRINT ALL CHARACTERS TO THE NEXT QUOTE.
2
3 016662  112237  002172          CON.QU: MOVB (R2)+,ERRCHR          ;GET CHARACTER
4 016666  123727  002172  000042  CMPB ERRCHR,#'"          ;CHECK IF ENDING QUOTE
5 016674  001413          BEQ CON.QX          ;IF SO, GO GET NEXT CONTROL CHARACTER
6 016676          PRINTX #ERRONE,#ERRCHR          ;PRINT THE CHARACTER
   016676  012746  002172          MOV          #ERRCHR,-(SP)
   016702  012746  005316          MOV          #ERRONE,-(SP)
   016706  012746  000002          MOV          #2,-(SP)
   016712  010600          MOV          SP,R0
   016714  104415          TRAP         C$PNTX
   016716  062706  000006          ADD          #6,SP
7 016722  000757          BR CON.QU          ;CONTINUE PRINTING
8 016724  000207          CON.QX: RETURN
9
10         ;CONTROL CHARACTER WAS AN A. PRINT ASCII CHARACTERS FROM PARAMETERS.
11
12 016726  004737  017504          CON.A: CALL GETCNT          ;GET COUNT OF CHARACTERS
13 016732  112437  002172          CON.A1: MOVB (R4)+,ERRCHR          ;GET CHARACTER
14 016736          PRINTX #ERRONE,#ERRCHR          ;PRINT THE CHARACTER
   016736  012746  002172          MOV          #ERRCHR,-(SP)
   016742  012746  005316          MOV          #ERRONE,-(SP)
   016746  012746  000002          MOV          #2,-(SP)
   016752  010600          MOV          SP,R0
   016754  104415          TRAP         C$PNTX
   016756  062706  000006          ADD          #6,SP
15 016762  005301          DEC R1          ;COUNT THE CHARACTERS
16 016764  001362          BNE CON.A1          ;PRINT UNTIL COUNT REACHES ZERO
17 016766  032704  000001          BIT #1,R4          ;CHECK IF R4 NOW ODD
18 016772  001401          BEQ CON.A2
19 016774  005204          INC R4          ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
20 016776  000207          CON.A2: RETURN          ;NOW GET NEXT CONTROL CHARACTER
21
22         ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.
23
24 017000  012701  000012          CON.D: MOV #10.,R1          ;LOAD RADIX
25 017004  004737  017562          CALL PNTNUM          ;PRINT NUMBER
26 017010  000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
27
28         ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.
29
30 017012  012701  000020          CON.H: MOV #16.,R1          ;LOAD RADIX
31 017016  004737  017562          CALL PNTNUM          ;PRINT NUMBER
32 017022  000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
33
34         ;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.
35
36 017024  012701  000010          CON.O: MOV #8.,R1          ;LOAD RADIX
37 017030  004737  017562          CALL PNTNUM          ;PRINT NUMBER
38 017034  000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
39
40         ;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.
41
42 017036  004737  017504          CON.N: CALL GETCNT          ;GET COUNT
43 017042          CON.N1: PRINTX #ERRNL          ;PRINT NEW LINE SEQUENCE
   017042  012746  005321          MOV          #ERRNL,-(SP)
   017046  012746  000001          MOV          #1,-(SP)

```

```

017052 010600
017054 104415
017056 062706 000004
44 017062 005301          DEC R1          ;COUNT THE SEQUENCES
45 017064 001366          BNE CON.N1
46 017066 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
47
48          ;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.
49
50 017070 004737 017504  CON.R:  CALL GETCNT          ;GET ROUTINE NUMBER
51 017074 020127 000002  CMP R1,#ERRRSZ          ;CHECK IF DEFINED ROUTINE NUMBER
52 017100 101004          BHI CON.R1
53 017102 060101          ADD R1,R1          ;DOUBLE COUNT TO GET WORD INDEX
54 017104 004771 017200  CALL @ERRRTB-2(R1)      ;CALL ROUTINE
55 017110 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
56 017112          CON.R1: PRINTF #ERRME1      ;REPORT BAD MESSAGE STRING
017112 012746 006332          MOV #ERRME1,-(SP)
017116 012746 000001          MOV #1,-(SP)
017122 010600          MOV SP,RO
017124 104417          TRAP C$PNTF
017126 062706 000004          ADD #4,SP
57 017132          POP R1          ;FIX THE STACK
017132 012601          MOV (SP)+,R1
58 017134 000207          RETURN
59
60          ;CONTROL CHARACTER WAS AN S. PRINT SPACES.
61
62 017136 004737 017504  CON.S:  CALL GETCNT          ;GET COUNT
63 017142 012737 000040 002172  MOV #' ',ERRCHR          ;STORE SPACE CHARACTER
64 017150          CON.S1: PRINTX #ERRONE,#ERRCHR ;PRINT THE SPACE
017150 012746 002172          MOV #ERRCHR,-(SP)
017154 012746 005316          MOV #ERRONE,-(SP)
017160 012746 000002          MOV #2,-(SP)
017164 010600          MOV SP,RO
017166 104415          TRAP C$PNTX
017170 062706 000006          ADD #6,SP
65 017174 005301          DEC R1          ;COUNT THE SPACES
66 017176 001364          BNE CON.S1
67 017200 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
68
69          ;ERROR ROUTINE DISPATCH TABLE
70
71 017202 017250  ERRRTB: .WORD CALR1          ;CALL ALTERNATE PRINT STRING
72 017204 017276          .WORD CALR2          ;PRINT AN SDI DIAGNOSE RESPONSE
73          ERRRSZ=<.-ERRRTB>/2
74
75 017206 002534  Tnames: .WORD T1NAME
76 017210 002556          .WORD T2NAME
77 017212 002574          .WORD T3NAME
78 017214 002612          .WORD T4NAME
79
80          ;BUILD TWO TABLES
81          ; FIRST CONTAINING CONTROL CHARACTERS
82          ; SECOND CONTAINING ROUTINE ADDRESSES
83
84          .MACRO BUILD
85          ENTRY "",CON.QU
    
```

```
86          ENTRY A,CON.A
87          ENTRY D,CON.D
88          ENTRY H,CON.H
89          ENTRY O,CON.O
90          ENTRY N,CON.N
91          ENTRY R,CON.R
92          ENTRY S,CON.S
93          .ENDM
94          ;HERE IS FIRST TABLE
95          .MACRO ENTRY ARG1,ARG2
96              .LIST
97              .BYTE 'ARG1
98              .NLIST
99          .ENDM
100
101
102
103 017216          ERRC:  BUILD
104 017216          .BYTE  ''
105 017217          .BYTE  'A
106 017220          .BYTE  'D
107 017221          .BYTE  'H
108 017222          .BYTE  'O
109 017223          .BYTE  'N
110 017224          .BYTE  'R
111 017225          .BYTE  'S
112 017226          .BYTE  0
113
114
115          ;FOLLOW WITH A NULL BYTE
          .EVEN
          ;HERE IS SECOND TABLE
          .MACRO ENTRY ARG1,ARG2
              .LIST
              .WORD ARG2
              .NLIST
          .ENDM
          ERRD:  BUILD
              .WORD CON.QU
              .WORD CON.A
              .WORD CON.D
              .WORD CON.H
              .WORD CON.O
              .WORD CON.N
              .WORD CON.R
              .WORD CON.S
```

```
1  
2  
3  
4 017250  
   017250 010246  
5 017252 012402  
6 017254 006302  
7 017256 063702 002212  
8 017262 067702 162724  
9 017266 004737 016602  
10 017272  
   017272 012602  
11 017274 000207
```

:PRE-PROGRAMMED ROUTINE 1
:CALL ALTERNATE PRINT STRING

CALR1: PUSH R2
 MOV R2,-(SP)
 MOV (R4)+,R2
 ASL R2
 ADD DMPROG,R2
 ADD @DMPROG,R2
 CALL OSTRNG
 POP R2
 MOV (SP)+,R2
 RETURN

:SAVE CURRENT STRING POINTER
:GET NEW STRING POINTER
:DOUBLE FOR WORD COUNT
:ADD START OF STRING STORAGE
:ADD SIZE OF MAIN PROGRAM
:OUTPUT USING THIS STRING
:GET OLD POINTER BACK
:NOW CONTINUE THE OLD STRING


```

1          ;PRE-PROGRAMMED ROUTINE 2
2          ;PRINT AN SDI DIAGNOSE RESPONSE
3
4 017276   CALR2:  PUSH R2                ;SAVE CURRENT STRING POINTER
5 017276   010246   MOV R2,-(SP)
6 017300   012402   MOV (R4)+,R2      ;GET COUNTS
7 017302   010246   PUSH R2                ;SAVE COUNTS
8 017302   000302   MOV R2,-(SP)
9 017304   000302   SWAB R2                ;GET BINARY COUNT
10 017306   042702 177400   BIC #177400,R2
11 017312   001420   BEQ 2$                ;BYPASS BINARY IF COUNT IS ZERO
12 017314   012700 000020   1$:  MOV #16.,R0        ;RADIX IS HEX
13 017320   012701 000040   MOV #32.,R1          ;32 BIT NUMBERS
14 017324   004737 017570   CALL PNTNUS          ;PRINT THE NUMBER
15 017330   012746 005321   PRINTX #ERRNL       ;GO TO NEW LINE
16 017334   012746 000001   MOV #ERRNL,-(SP)
17 017340   010600   MOV #1,-(SP)
18 017342   104415   MOV SP,R0
19 017344   062706 000004   TRAP C$PNTX
20 017350   005302   ADD #4,SP
21 017352   001360   DEC R2
22 017354   012601   BNE 1$
23 017354   012601   2$:  POP R1                ;GET ASCII COUNT
24 017354   012601   MOV (SP)+,R1
25 017356   042701 177400   BIC #177400,R1
26 017362   001412   BEQ 3$                ;BYPASS IS COUNT IS ZERO
27 017364   004737 016732   CALL CON.A1          ;PRINT THE ASCII
28 017370   012746 005321   PRINTX #ERRNL       ;GO TO NEW LINE
29 017374   012746 000001   MOV #ERRNL,-(SP)
30 017400   010600   MOV #1,-(SP)
31 017402   104415   MOV SP,R0
32 017404   062706 000004   TRAP C$PNTX
33 017410   012602   ADD #4,SP
34 017412   000207   3$:  POP R2                ;RESTORE STRING POINTER
35 017412   000207   MOV (SP)+,R2
36 017412   000207   RETURN
    
```

```

1      :GTDRVT
2
3      :GET DRIVE TABLE POINTER
4
5      :INPUTS:
6          R5 - CONTROLLER TABLE ADDRESS
7          R1 - DRIVE NUMBER
8
9      :OUTPUTS:
10         R4 - DRIVE TABLE ADDRESS
11         L$LUN - LOADED WITH UNIT NUMBER OF DRIVE
12         Z CLEAR IF DRIVE TABLE NOT FOUND AFTER ERROR PRINTED
13
14 017414 010246
15 017416 010504
16 017420 062704 000020
17 017424 012702 000010
18 017432 001406
19 017434 027401 000000
20 017440 001412
21 017442 005724
22 017444 005302
23 017446 001370
24 017450
    017450 104455
    017452 000120
    017454 004547
    017456 000000
25 017460
    017460 012602
26 017462 000244
27 017464 000207
28 017466 011404
29 017470 016437 000002 002074
30 017476
    017476 012602
31 017500 000264
32 017502 000207

    GTDRVT: PUSH R2
                MOV R2,-(SP)
    MOV R5,R4           ;GET CONTROLLER TABLE ADDRESS
    ADD #C.DR0,R4      ;ADD OFFSET TO DRIVE TABLE ADDRESS
    MOV #8,R2          ;GET COUNT OF DRIVES
    1$: TST (R4)        ;CHECK IF AN ADDRESS HERE
        BEQ 3$
        CMP @ (R4),R1  ;COMPARE DRIVE NUMBERS
        BEQ 10$        ;BRANCH IF A MATCH
    2$: TST (R4)+      ;BUMP ADDRESS
        DEC R2
        BNE 1$         ;LOOK AT ALL OF THEM
    3$: ERRDF 80,GTDRV1 ;UNIT NUMBER NOT FOUND
                                TRAP C$ERDF
                                .WORD 80
                                .WORD GTDRV1
                                .WORD 0
    POP R2
    MOV (SP)+,R2
    CLZ                ;CLEAR Z AS ERROR FLAG
    RETURN
    10$: MOV (R4),R4   ;GET ADDRESS OF TABLE
        MOV D.UNIT(R4),L$LUN ;GET UNIT NUMBER
        POP R2
        MOV (SP)+,R2
    SEZ                ;SET Z FLAG
    RETURN
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13 017504
14 017504 010046
15 017506 005001
16 017510 121227 000060
17 017514 103415
18 017516 121227 000071
19 017522 101012
20 017524 006301
21 017526 010100
22 017530 006301
23 017532 006301
24 017534 060001
25 017536 112200
26 017540 162700 000060
27 017544 060001
28 017546 000760
29 017550 005701
30 017552 001001
31 017554 005201
32 017556 012600
33 017560 060207

```

:GETCNT
:
:GET COUNT IN NEXT CHARACTERS OF STRING POINTED TO BY R2.
:NUMBER WILL BE IN DECIMAL. IF NO NUMBER, RETURN A
:DEFAULT OF 1.
:
:INPUTS:
:   R2 - POINTER TO ASCII STRING
:OUTPUTS:
:   R1 - NUMBER READ OR A ONE
:   R2 - POINTING TO CHARACTER AFTER NUMBER
:
GETCNT: PUSH R0
        MOV R0,-(SP)
        CLR R1
        :START WITH ZERO COUNT
GETCNX: CMPB (R2),#'0
        BLO GETCDN
        :CHECK IF CHARACTER A DIGIT
        :BRANCH IF LOWER THAN ZERO
        CMPB (R2),#'9
        BHI GETCDN
        :BRANCH IF HIGHER THAN NINE
        ASL R1
        :MULTIPLY NUMBER BY 10
        MOV R1,R0
        :SAVE 2N
        ASL R1
        :COMPUTE 4N
        ASL R1
        :COMPUTE 8N
        ADD R0,R1
        :8N + 2N = 10N
        MOVB (R2)+,R0
        :GET DIGIT FROM STING
        SUB #'0,R0
        :GET RID OF ASCII
        ADD R0,R1
        :ADD TO NUMBER
        BR GETCNX
        :GO TO NEXT CHARACTER
GETCDN: TST R1
        BNE GETCXX
        :CHECK IF NUMBER IS ZERO
        :IF ZERO, CHANGE
        INC R1
        : TO DEFAULT OF ONE
GETCXX: POP R0
        MOV (SP)+,R0
        RETURN
    
```

1			:PNTNUM	
2			:PRINT A NUMBER	
3			:INPUTS:	
4			R1 - RADIX OF NUMBER	
5			R2 - ASCII STRING TO COUNT OF BITS IN NUMBER	
6			R4 - POINTER TO NUMBER (LOW WORD)	
7			:OUTPUTS:	
8			NUMBER IS PRINTED. LEADING ZEROS ARE PRINTED EXCEPT FOR	
9			DECIMAL NUMBERS.	
10			R0 - CONTENTS DESTROYED	
11				
12				
13				
14	017562	010100	PNTNUM: MOV R1,R0	:SAVE RADIX
15	017564	004737 017504	CALL GETCNT	:GET COUNT OF BITS
16	017570		PNTNUS: PUSH <R2,R3,R5>	
	017570	010246	MOV R2,-(SP)	
	017572	010346	MOV R3,-(SP)	
	017574	010546	MOV R5,-(SP)	
17	017576	012403	MOV (R4)+,R3	:GET ONE PARAMETER WORD
18	017600	005005	CLR R5	:CLEAR STORAGE FOR OTHER
19	017602	020127 000020	CMP R1,#16.	:MORE THAN 16 BITS IN NUMBER?
20	017606	003401	BLE 1\$	
21	017610	012405	MOV (R4)+,R5	:YES, GET SECOND PARAMETER WORD
22	017612		1\$: PUSH R4	
	017612	010446	MOV R4,-(SP)	
23	017614	010504	MOV R5,R4	:PUT HIGH WORD IN R4
24	017616	012702 000020	MOV #16.,R2	:COMPUTE BITS NOT WANTED
25	017622	160102	SUB R1,R2	:BY SUBTRACTING BITS TO USE
26	017624	002002	BGE 2\$:FROM 16.
27	017626	062702 000020	ADD #16.,R2	:IF NEGATIVE, ADD 16 FOR FIRST WORD
28	017632	001414	2\$: BEQ 6\$:IF ZERO, NO BITS NEED BE CLEARED
29	017634	012705 100000	MOV #BIT15,R5	:START MASK WITH SIGN BIT SET
30	017640	005302	3\$: DEC R2	:COUNT BITS IN MASK
31	017642	001402	BEQ 4\$	
32	017644	006205	ASR R5	:SHIFT MORE BITS TO RIGHT
33	017646	000774	BR 3\$	
34	017650	020127 000020	4\$: CMP R1,#16.	:MORE THAN 16 BITS IN NUMBER?
35	017654	003402	BLE 5\$	
36	017656	040504	BIC R5,R4	:YES, CLEAR IN HIGH WORD
37	017660	000401	BR 6\$	
38	017662	040503	5\$: BIC R5,R3	:NO, CLEAR IN LOW WORD
39	017664	004737 020070	6\$: CALL DIVIDE	:DIVIDE BY RADIX IN R0
40	017670		PUSH R5	:PUSH REMAINDER ON STACK
	017670	010546	MOV R5,-(SP)	
41	017672	005202	INC R2	:COUNT DIGITS ON STACK
42	017674	005703	TST R3	:CHECK IF QUOTIENT IS ZERO
43	017676	001372	BNE 6\$	
44	017700	005704	TST R4	
45	017702	001370	BNE 6\$	
46	017704	020027 000012	CMP R0,#10.	:IF RADIX IS DECIMAL
47	017710	001434	BEQ 10\$: JUST GO PRINT DIGITS ON STACK
48	017712	010103	MOV R1,R3	:OTHERWISE COMPUTE NUMBER OF LEADING ZEROS
49	017714	162700 000014	SUB #12.,R0	:DIVIDEND IS BITS IN NUMBER
50	017720	003002	BGT 7\$:DIVISOR IS BITS PER DIGIT PRINTED
51	017722	012700 000003	MOV #3,R0	: (3 OR 4)
52	017726	004737 020070	7\$: CALL DIVIDE	

```

53 017732 005705          TST R5          ;IF REMAINDER NOT ZERO
54 017734 001401          BEQ 8$          ;INCREMENT QUOTIENT
55 017736 005203          INC R3
56 017740 160203          8$: SUB R2,R3      ;SUBTRACT DIGITS ON STACK
57 017742 001417          BEQ 10$         ;NO LEADING ZEROS IF ZERO
58 017744 112737 000060 002172  MOVB #'0,ERRCHR ;STORE ZERO CHARACTER
59 017752          9$: PRINTX #ERRONE,#ERRCHR ;PRINT A ZERO
    017752 012746 002172          MOV #ERRCHR,-(SP)
    017756 012746 005316          MOV #ERRONE,-(SP)
    017762 012746 000002          MOV #2,-(SP)
    017766 010600          MOV SP,R0
    017770 104415          TRAP C$PNTX
    017772 062706 000006          ADD #6,SP
60 017776 005303          DEC R3
61 020000 001364          BNE 9$          ;REPEAT UNTIL COUNT REACHES ZERO
62
63 020002          10$: POP R5          ;GET CHACACTER FROM STACK
    020002 012605          MOV (SP)+,R5
64 020004 062705 000060          ADD #'0,R5      ;CNVERT TO ASCII DIGIT
65 020010 020527 000071          CMP R5,#'9     ;IF GREATER THAN A 9
66 020014 003402          BLE 11$       ; CONVERT TO A OR HIGHER
67 020016 062705 000007          ADD #'A-'9-1>,R5 ; FOR HEX DIGIT
68 020022 110537 002172          11$: MOV R5,ERRCHR ;STORE CHARACTER
69 020026          PRINTX #ERRONE,#ERRCHR ;PRINT IT
    020026 012746 002172          MOV #ERRCHR,-(SP)
    020032 012746 005316          MOV #ERRONE,-(SP)
    020036 012746 000002          MOV #2,-(SP)
    020042 010600          MOV SP,R0
    020044 104415          TRAP C$PNTX
    020046 062706 000006          ADD #6,SP
70 020052 005302          DEC R2          ;REPEAT FOR ALL DIGITS
71 020054 001352          BNE 10$        ; ON STACK
72 020056          POP <R4,R5,R3,R2>
    020056 012604          MOV (SP)+,R4
    020060 012605          MOV (SP)+,R5
    020062 012603          MOV (SP)+,R3
    020064 012602          MOV (SP)+,R2
73 020066 000207          RETURN
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 020070
020070 010246
17 020072 012702 000040
18 020076 005005
19 020100 006303
20 020102 006104
21 020104 006105
22 020106 020005
23 020110 101002
24 020112 160005
25 020114 005203
26 020116 005302
27 020120 001367
28 020122
020122 012602
29 020124 000207

```

:DIVIDE
:DIVIDE A 32 BIT UNSIGNED NUMBER BY A 16 BIT UNSIGNED NUMBER.
:REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
:WILL NOT CHECK FOR DIVIDE BY ZERO.
:INPUTS:
R3 - LOW 16 BITS OF DIVIDEND
R4 - HIGH 16 BITS OF DIVIDEND
R0 - DIVISOR
:OUTPUTS:
R3 - LOW 16 BITS OF QUOTIENT
R4 - HIGH 16 BITS OF QUOTIENT
R5 - REMAINDER

DIVIDE: PUSH R2
MOV R2,-(SP)
MOV #32,,R2 ;SET UP SHIFT COUNT
CLR R5 ;START WITH ZERO REMAINDER
1$: ASL R3 ;SHIFT LEFT INTO R5
ROL R4
ROL R5
CMP R0,R5 ;WILL DIVISOR GO INTO REMAINDER
BHI 2$ ;ONLY SUBTRACT IF IT WILL
SUB R0,R5 ;SUBTRACT DIVISOR
INC R3 ;PUT A ONE INTO QUOTIENT
2$: DEC R2 ;COUNT THE SHIFTS
BNE 1$
POP R2
MOV (SP)+,R2
RETURN
    
```

```

1      ;BUILD DEFAULT 28-BIT NUMBER
2
3      ;INPUT:
4      R4 - POINTER TO 2 WORD DEFAULT NUMBER
5
6      ;OUTPUT:
7      TEMP - ASCIZ STRING REPRESENTING DEFAULT NUMBER
8 020126 BLD28: PUSH <R0,R1,R3,R4,R5>
    020126          MOV R0,-(SP)
    020130          MOV R1,-(SP)
    020132          MOV R3,-(SP)
    020134          MOV R4,-(SP)
    020136          MOV R5,-(SP)
9 020140          MOV (R4),R3          ;GET NUMBER
10 020142          MOV 2(R4),R4
11 020146          MOV #10.,R0        ;DIVISOR IS 10.
12 020152          CLR R1            ;CLEAR CHARACTER COUNT
13 020154          1$: CALL DIVIDE
14 020160          ADD #'0.,R5        ;CONVERT REMAINDER TO ASCII CHARACTER
15 020164          PUSH R5           ;STORE ON STACK
    020164          MOV R5,-(SP)
16 020166          INC R1            ;COUNT THE CHARACTER
17 020170          MOV R3,R5          ;REPEAT UNTIL QUOTIENT IS ZERO
18 020172          BIS R4,R5
19 020174          BNE 1$
20 020176          MOV #TEMP,R0      ;GET POINTER TO STRING
21 020202          2$: POP R5        ;PUT CHARACTERS INTO STRING
    020202          MOV (SP)+,R5
22 020204          MOVB R5,(R0)+
23 020206          DEC R1
24 020210          BNE 2$
25 020212          CLRB (R0)+
26 020214          POP <R5,R4,R3,R1,R0> ;END WITH NULL
    020214          MOV (SP)+,R5
    020216          MOV (SP)+,R4
    020220          MOV (SP)+,R3
    020222          MOV (SP)+,R1
    020224          MOV (SP)+,R0
27 020226          RETURN
    000002
    000012
    020070
    000060
    010046
    010146
    010346
    010446
    010546
    011403
    016404
    012700
    005001
    004737
    062705
    010546
    005201
    010305
    050405
    001367
    012700
    002260
    012605
    110520
    005301
    001374
    105020
    012605
    012604
    012603
    012601
    012600
    000207
    
```

```

1          : CONVERT ASCII STRING TO 28-BIT NUMBER
2
3          : INPUTS:
4          : TEMP - ASCII STRING UP TO 9 CHARACTERS LONG
5          : R4 - ADDRESS OF TWO WORD STORAGE
6
7          : OUTPUTS:
8          : IF STRING IS VALID NUMBER
9          : TWO WORDS AT R4 LOADED WITH NUMBER
10         : R4 POINTING TO WORD AFTER STORAGE
11         : CARRY CLEAR
12         : IF STRING INVALID
13         : ERROR MESSAGE PRINTED
14         : CARRY SET
15
16 020230 010046 CNV28: PUSH <R0,R1,R2,R3>
17 020232 010146          MOV R0,-(SP)
18 020234 010246          MOV R1,-(SP)
19 020236 010346          MOV R2,-(SP)
20 020240 005000          MOV R3,-(SP)
21 020242 005001          CLR R0
22 020244 012702 002260 1$: MOV #TEMP,R2
23 020250 112203          MOV B(R2)+,R3
24 020252 001452          BEQ 3$
25 020254 162703 000060  SUB #'0,R3
26 020260 100435          BMI 2$
27 020262 022703 000011  CMP #'9.,R3
28 020266 103432          BLO 2$
29 020270 006300          ASL R0
30 020272 006101          ROL R1
31 020274          PUSH <R1,R0>
32 020276 010146          MOV R1,-(SP)
33 020278 010046          MOV R0,-(SP)
34 020300 006300          ASL R0
35 020302 006101          ROL R1
36 020304 006300          ASL R0
37 020306 006101          ROL R1
38 020310 062600          ADD (SP)+,R0
39 020312 005501          ADC R1
40 020314 062601          ADD (SP)+,R1
41 020316 060300          ADD R3,R0
42 020320 005501          ADC R1
43 020322 032701 170000  BIT #170000,R1
44 020324 001750          BEQ 1$
45 020330          PRINTF #INP28A
46 020332 012746 010002  MOV #INP28A,-(SP)
47 020334 012746 000001  MOV #1,-(SP)
48 020340 010600          MOV SP,R0
49 020342 104417          TRAP C$PNTF
50 020344 062706 000004  ADD #4,SP
51 020350 000261          SEC
52 020352 000415          BR 4$
53
54 020354          2$: PRINTF #INP28B
55 020356 012746 010044  MOV #INP28B,-(SP)
56 020360 012746 000001  MOV #1,-(SP)
57 020364 010600          MOV SP,R0

```


				TRAP	C\$PNTF
				ADD	#4,SP
020366	104417				
020370	062706	000004			
44 020374	000261		SEC		
45 020376	000403		BR 4\$		
46					
47 020400	010024	3\$:	MOV R0,(R4)+		;MOVE NUMBER TO STORAGE AREA
48 020402	010124		MOV R1,(R4)+		
49 020404	000241		CLC		;CLEAR CARRY TO INDICATE ALL IS WELL
50 020406		4\$:	POP <R3,R2,R1,R0>		
020406	012603		MOV (SP)+,R3		
020410	012602		MOV (SP)+,R2		
020412	012601		MOV (SP)+,R1		
020414	012600		MOV (SP)+,R0		
51 020416	000207		RETURN		

```

1
2
3
4 020420 112737 000040 002172 T2PNTW: MOV B #' ,ERRCHR          ;PRINT A SPACE
5 020426 012746 002172          PRINTF #ERRONE,#ERRCHR
6 020426 012746 002172          MOV #ERRCHR,-(SP)
7 020432 012746 005316          MOV #ERRONE,-(SP)
8 020436 012746 000002          MOV #2,-(SP)
9 020442 010600          MOV SP,R0
10 020444 104417          TRAP C$PNTF
11 020446 062706 000006          ADD #6,SP
12 020452          PUSH R1
13 020452 010146          MOV R1,-(SP)
14 020454 000301          SWAB R1
15 020456 004737 020524          CALL T2PNT          ;PRINT HIGH TWO DIGITS
16 020462          POP R1
17 020462 012601          MOV (SP)+,R1
18 020464 004737 020524          CALL T2PNT          ;PRINT LOW TWO DIGITS
19 020470 000207          RETURN
20
21
22 020472 112737 000040 002172 T2PNTB: MOV B #' ,ERRCHR          ;PRINT A SPACE
23 020500 012746 002172          PRINTF #ERRONE,#ERRCHR
24 020500 012746 002172          MOV #ERRCHR,-(SP)
25 020504 012746 005316          MOV #ERRONE,-(SP)
26 020510 012746 000002          MOV #2,-(SP)
27 020514 010600          MOV SP,R0
28 020516 104417          TRAP C$PNTF
29 020520 062706 000006          ADD #6,SP
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

1      ;T2GNUM
2
3      ;GET A HEX DIGIT FROM AN ASCII INPUT STRING
4
5      ;INPUTS:
6      ;R1 - STRING POINTER
7      ;OUTPUTS:
8      ;R4 - NUMBER
9      ;R1 - UPDATED STRING TO CHARACTER AFTER NUMBER
10     ;R0 - COUNT OF DIGITS (0 IF END OF LINE FOUND)
11
12 020620 005000      T2GNUM: CLR R0          ;CLEAR DIGIT COUNT
13 020622 105711      T2GNUM: TSTB (R1)       ;CHECK IF END OF LINE
14 020624 001442      T2GNUM: BEQ T2GNX        ;REPORT NULL CHARACTER FOUND
15 020626 121127 000040 T2GNUM: CMPB (R1),#' ' ;CHECK IF A SPACE
16 020632 001002      T2GNUM: BNE T2GND1      ;IF SO, IGNORE IT
17 020634 005201      T2GNUM: INC R1
18 020636 000770      T2GNUM: BR T2GNUM
19 020640 005004      T2GND1: CLR R4          ;CLEAR NUMBER STORAGE
20 020642 000000      T2GND2: PUSH R2         ;SAVE REGISTER
21 020644 112102      T2GND2: MOV R2,-(SP)
22 020646 162702 000060 T2GND2: MOVB (R1)+,R2      ;GET CHARACTER
23 020652 100431      T2GND2: SUB #'0,R2      ;CONVERT TO HEX DIGIT
24 020654 020227 000011 T2GND2: BMI T2GNE
25 020660 003410      T2GND2: CMP R2,#9
26 020662 020227 000021 T2GND2: BLE T2GND3
27 020666 103423      T2GND2: CMP R2,#<'A-'0>
28 020670 020227 000026 T2GND2: BLO T2GNE
29 020674 101020      T2GND2: CMP R2,#<'F-'0>
30 020676 162702 000007 T2GND2: BHI T2GNE
31 020702 006304      T2GND3: SUB #'A-'9-1>,R2
32 020704 006304      T2GND3: ASL R4
33 020706 006304      T2GND3: ASL R4
34 020710 006304      T2GND3: ASL R4
35 020712 050204      T2GND3: ASL R4
36 020714 005200      T2GND3: BIS R2,R4
37 020716 012602      T2GND3: INC R0
38 020720 105711      T2GND3: POP R2
39 020722 001403      T2GND3: MOV (SP)+,R2
40 020724 121127 000040 T2GND3: TSTB (R1)
41 020730 001344      T2GND3: BEQ T2GNX
42 020732 005700      T2GND3: CMPB (R1),#' '
43 020734 000207      T2GND3: BNE T2GND2
44
45 020736 012602      T2GNX: TST R0
46 020740 012600      T2GNX: RETURN
47 020742 000137 015126 T2GNE: POP R2          ;CLEAN UP THE STACK
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

1      :PNTERR
2
3      :PRINT ERROR MESSAGE FROM DM PROGRAM REQUEST 11 OR 12.
4
5      :INPUTS:
6          R5 - CONTROLLER TABLE ADDRESS
7          R4 - MESSAGE DATA ADDRESS
8      :OUTPUTS:
9          NONE
10
11     020746      PNTERR: PUSH <R0,R1,R2,R3>
12     020746      010046          MOV R0,-(SP)
13     020750      010146          MOV R1,-(SP)
14     020752      010246          MOV R2,-(SP)
15     020754      010346          MOV R3,-(SP)
16     020756      005764 000004      TST 4(R4)          ;GET DRIVE NUMBER
17     020762      002004          BGE 1$            ;CHECK IF BIT 15 SET
18     020764      016537 000002 002074  MOV C.UNIT(R5),L$LUN ;IF SO, GET UNIT FROM CONTROLLER TABLE
19     020772      000406          BR 2$
20     020774      010446      1$: PUSH R4          ;SAVE DATA ADDRESS
21     020774      016401          MOV R4,-(SP)
22     021002      004737 017414      MOV 4(R4),R1      ;GET DRIVE NUMBER
23     021006          CALL GTDRVT      ;GET DRIVE TABLE ADDRESS
24     021006          POP R4          ;RESTORE DATA ADDRESS
25     021010      012702 002162      2$: MOV #ERRRTP,R2      ;GET POINTER TO ERROR TABLE
26     021014      116422 000003      MOV 3(R4),(R2)+  ;GET ERROR TYPE
27     021020      105022          CLR 2(R2)+       ;CLEAR HIGH BYTE
28     021022      116422 000002      MOV 2(R4),(R2)+  ;GET ERROR NUMBER
29     021026      105022          CLR 2(R2)+       ;CLEAR HIGH BYTE
30     021030      005022          CLR (R2)+        ;CLEAR MESSAGE POINTER
31     021032      012712 021052      MOV #ERRRTN,(R2) ;GET ROUTINE NUMBER
32     021036          ERROR          ;PRINT THE ERROR MESSAGE
33     021040          POP <R3,R2,R1,R0>
34     021040      012603          MOV (SP)+,R3
35     021042      012602          MOV (SP)+,R2
36     021044      012601          MOV (SP)+,R1
37     021046      012600          MOV (SP)+,R0
38     021050      000207          RETURN
39
40     BGNMSG ERRRTN          ;ERROR REPORT ROUTINE
41     021052          ERRRTN::
42     021052      013702 002226      MOV TNUM,R2      ;GET TEST NUMBER
43     021056      006302          ASL R2          ;DOUBLE
44     021060          PRINTB #ERRMB,TNAMES-2(R2),(R4),C.UADR(R5) ;PRINT BASIC MESSAGE LINE WITH DM PC
45     021060      016546 000000          MOV C.UADR(R5),-(SP)
46     021064      011446          MOV (R4),-(SP)
47     021066      016246 017204          MOV TNAMES-2(R2),-(SP)
48     021072      012746 006216          MOV #ERRMB,-(SP)
49     021076      012746 000004          MOV #4,-(SP)
50     021102      010600          MOV SP,R0
51     021104      104414          TRAP C$PNTB
52     021106      062706 000012          ADD #12,SP
53     021112      005764 000004      TST 4(R4)          ;CHECK IF DRIVE NUMBER GIVEN
54     021116      100412          BMI 1$          ;BRANCH IF NOT
55     021120          PRINTB #ERRMBD,4(R4) ;PRINT DRIVE NUMBER

```

021120 016446 000004
 021124 012746 006260
 021130 012746 000002
 021134 010600
 021136 104414
 021140 062706 000006
 38 021144 004737 024436
 39 021150 062704 000006
 40 021154 012402
 41 021156 006302
 42 021160 063702 002212
 43 021164 067702 161022
 44 021170 105712
 45 021172 001001
 46 021174 005202
 47 021176 004737 016602
 48 021202
 021202
 021202 104423
 49

1\$: CALL RNTIME
 ADD #6,R4
 MOV (R4)+,R2
 ASL R2
 ADD DMPROG,R2
 ADD @DMPROG,R2
 TSTB (R2)
 BNE NCON
 INC R2
 NCON: CALL OSTRNG
 ENDMSG

:PRINT RUNTIME
 :INCREASE R4 TO POINT TO MESSAGE POINTER
 :GET MESSAGE POINTER
 :DOUBLE TO MAKE BYTE OFFSET
 :ADD TO START OF MESSAGE STRINGS
 :ADD SIZE OF MAIN PROGRAM
 :CHECK FIRST BYTE
 :IF ZERO
 : INCREMENT TO NEXT BYTE
 :OUTPUT ACCORDING TO STRING

MOV 4(R4),-(SP)
 MOV #ERRMBD,-(SP)
 MOV #2,-(SP)
 MOV SP,R0
 TRAP C\$PNTB
 ADD #6,SP

L10020:
 TRAP C\$MSG

```

1      ;LOADDM
2      ;
3      ;LOAD AND START A DM PROGRAM INTO A CONTROLLER
4      ;
5      ;INPUTS:
6      ;   R5 - CONTROLLER TABLE ADDRESS
7      ;
8      ;IMPLICIT INPUTS:
9      ;   DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
10     ;
11     ;OUTPUTS:
12     ;   IF LOAD SUCCEEDS - Z CLEAR
13     ;           CONTROLLER TABLE MARKED LOADED
14     ;
15     ;   IF ERROR - Z SET
16
17     LOADDM:
18     MOV C.VEC(R5),R4           ;GET VECTOR OF UDA
19     AND CT.VEC,R4
20     BIC #^C<CT.VEC>,R4
21     MOV R5,R1                 ;GET INTERRUPT SERVICE LINK
22     ADD #C.JSR,R1
23     SETVEC R4,R1,#PRI07      ;SET UP INTERRUPT VECTOR
24
25     MOV #PRI07,-(SP)
26     MOV R1,-(SP)
27     MOV R4,-(SP)
28     MOV #3,-(SP)
29     TRAP C$$VEC
30     ADD #10,SP
31
32     ;INITIALIZE UDA WITH SMALLEST
33     ;POSITION VECTOR FOR UDA
34     ASR R4
35     ASR R4
36     CALL UDAINIT             ; RING BUFFER AND INTERRUPTS ENABLED
37     BEQ LOADER               ; BRANCH IF AN ERROR
38     CALL HCOMM               ; ALLOCATE SPACE FOR HOST COMM AREA
39
40     ;IF TEST NUMBER 1
41     ; DO SPECIAL LOAD
42     ; GET SIZE OF PROGRAM
43     ; BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET
44     ;
45     ;LOAD MAIN PROGRAM ADDRESS
46     ; AND SIZE
47     ;LOAD OVERLAY ADDRESS
48
49     LOADB:
50     CMP TNUM,#1
51     BEQ LOADT1
52     MOV @DMPROG,R1
53     MOV #OP.ESP,R0
54     CALL BLDCMD
55     MOV DMPROG,HC.CPK+P.UADR(R4)
56     MOV R1,HC.CPK+P.BCNT(R4)
57     MOV DMPROG,HC.CPK+P.OVRL(R4)
58     ADD @DMPROG,HC.CPK+P.OVRL(R4)
59     CALL SNDCMD
60     BEQ LOADER
61     CALL WAITMS
62     BIT #ST.MSK,HC.MPK+P.STS(R4)
63     BNE LOADE1
64     BIC #CT.CMD+CT.REQ,C.FLG(R5) ;CLEAR COMMAND OUTSTANDING FLAG
65     BIS #CT.RN,C.FLG(R5)        ;SET DM PROGRAM RUNNING FLAG
66     RETURN

```



```

1      ;LOAD
2      ;
3      ;ISSUE DOWNLINE LOAD COMMAND TO UDA. CHECK THAT LOAD
4      ;HAPPENS WITHOUT ERROR.
5      ;
6      ;INPUTS:
7      ;
8      ;   R1 - OFFSET FOR DM PROGRAM
9      ;   R2 - ADDRESS OF BUFFER CONTAINING PROGRAM
10     ;   R3 - SIZE OF BUFFER IN BYTES
11     ;   R5 - CONTROLLER TABLE ADDRESS
12     ;
13     ;OUTPUTS:
14     ;   Z CLEAR IF NO ERROR
15     ;   Z SET IF ERROR AND ERROR REPORTED
16     LOAD:  PUSH <R0,R3,R4>
17             MOV R0,-(SP)
18             MOV R3,-(SP)
19             MOV R4,-(SP)
20             MOV #OP.MWR,R0           ;GET DOWNLINE LOAD COMMAND
21             CALL BLDCMD             ;BUILD COMMAND PACKET
22             MOV R2,HC.CPK+P.UADR(R4) ;STUFF IN BUFFER ADDRESS
23             MOV R3,HC.CPK+P.BCNT(R4) ;STUFF IN BYTE COUNT
24             MOV R1,HC.CPK+P.RGOF(R4) ;STUFF IN OFFSET
25             MOV #1,HC.CPK+P.RGID(R4) ;STUFF IN REGION ID 1
26             CALL SND CMD           ;SEND COMMAND TO UDA
27             BEQ LOADER
28             CALL WAITMS            ;WAIT FOR MESSAGE RESPONSE
29             BIT #ST.MSK,HC.MPK+P.STS(R4) ;LOOK FOR ANY ERROR
30             BNE LOADE1
31             BIC #CT.CMD,C.FLG(R5)   ;CLEAR COMMAND ISSUED
32             POP <R4,R3,R0>
33             MOV (SP)+,R4
34             MOV (SP)+,R3
35             MOV (SP)+,R0
36
37             CLZ                       ;CLEAR Z TO INDICATE NO ERROR
38             RETURN
    
```

```

15 021506
   021506 010046
   021510 010346
   021512 010446
16 021514 012700 000031
17 021520 004737 021624
18 021524 010264 000124
19 021530 010364 000120
20 021534 010164 000144
21 021540 012764 000001 000140
22 021546 004737 021710
23 021552 001422
24 021554 004737 022050
25 021560 032764 000037 000032
26 021566 001010
27 021570 042765 000004 000014
28 021576
   021576 012604
   021600 012603
   021602 012600
29 021604 000244
30 021606 000207
    
```



```
1 ;UDA FAILED TO DOWNLINE LOAD DM PROGRAM
2
3 021610
  021610 104456
  021612 000024
  021614 003206
  021616 000000
4 021620 000264
5 021622 000207

LOADER: SEZ
        RETURN

LOADE1: ERRHRD 20,LOADM1

TRAP   C$ERHRD
.WORD  20
.WORD  LOADM1
.WORD  0

;SET Z TO INDICATE ERROR OCCURRED
```

```

1      :BLDCMD
2
3      :BUILD A COMMAND IN COMMAND PACKET
4
5      :INPUTS:
6          R5 - CONTROLLER TABLE ADDRESS
7          R0 - COMMAND CODE
8
9      :OUTPUTS:
10         R4 - ADDRESS OF HOST COMM AREA
11         COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL OTHER FIELDS CLEARED.
12         CMD REFERENCE NUMBER IN CONTROLLER TABLE INCREMENTED AND RESULT
13         IN COMMAND PACKET.
14         R0 - CONTENTS DESTROYED
15
16 021624 010146
17 021624 010046
18 021630 016504 000016
19 021634 010400
20 021636 062700 000100
21 021642 012720 000060
22 021646 012701 001000
23 021652 022716 000031
24 021656 001002
25 021660 012701 177777
26 021664 010120
27 021666 012701 000030
28 021672 005020
29 021674 005301
30 021700 012664 000114
31 021704 012601
32 021706 000207

BLDCMD: PUSH <R1,R0>
        MOV R1,-(SP)
        MOV R0,-(SP)
        MOV C.RING(R5),R4          :GET ADDRESS OF HOST COMM AREA
        MOV R4,R0                 :COPY TO R0
        ADD #HC.CEV,R0            :COMPUTE ADDRESS OF COMMAND ENVELOPE
        MOV #HC.PSZ,(R0)+        :LOAD PACKET LENGTH
        MOV #DUP,R1              :LOAD DIAG CIRCUIT IDENTIFIER
        CMP #OP.MWR,(SP)        :IF CODE IS MAINTENANCE WRITE
        BNE BLDC0                : GET OTHER CIRCUIT IDENTIFIER
        MOV #DIAG,R1
BLDC0:  MOV R1,(R0)+             :PUT IDENTIFIER INTO PACKET
        MOV #<HC.PSZ>/2,R1      :GET WORDS TO CLEAR
BLDC1:  CLR (R0)+                :CLEAR PACKET
        DEC R1
        BNE BLDC1
        POP HC.CPK+P.OPCD(R4)    :PUT OPCODE IN PACKET
        MOV (SP)+,HC.CPK+P.OPCD(R4)
        POP R1                   :RESTORE R1
        MOV (SP)+,R1
        RETURN
    
```

```

1      :SNDCMD
2      :
3      :SEND A COMMAND TO THE UDA.
4      :CLEAR THE RESPONSE PACKET. MARK BOTH PACKETS AVAILABLE TO THE
5      :UDA. SET COMMAND ISSUED BIT IN CONTROLLER TABLE AND INITIALIZE
6      :TIMEOUT COUNTER.
7      :
8      :INPUTS:
9      :   R5 - CONTROLLER TABLE ADDRESS
10     :
11     :OUTPUTS:
12     :   R4 - ADDRESS OF HOST COMM AREA
13     :
14     SNDCMD: PUSH <R0,R1>
15     021710 010046          MOV R0,-(SP)
16     021712 010146          MOV R1,-(SP)
17     021714 016504 000016   MOV C.RING(R5),R4          :LOAD R4 WITH HOST COMM AREA ADDRESS
18     021720 005265 000044   INC C.REF(R5)             :INCREMENT CMD REFERENCE NUMBER
19     021724 016564 000044 000104 MOV C.REF(R5),HC.CPK+P.CRF(R4) :PUT IN PACKET
20     021732 012700 000014   MOV #HC.MEV,R0           :POINT TO MESSAGE ENVELOPE
21     021736 060400          ADD R4,R0
22     021740 012701 000032   MOV #<HC.PSZ+HC.ESZ>/2,R1 :SIZE OF MESSAGE PACKET
23     021744 005020          SNDC1: CLR (R0)+          :CLEAR ENTIRE MESSAGE PACKET
24     021746 005301          DEC R1
25     021750 001375          BNE SNDC1
26     021752 012764 140000 000006 MOV #RG.OWN+RG.FLG,HC.MCT(R4) :MARK MESSAGE PACKET AVAILABLE
27     021760 012764 100000 000012 MOV #RG.OWN,HC.CCT(R4)       :MARK COMMAND TO UDA
28     021766 005775 000000          TST @ (R5)                :TELL UDA COMMAND IS THERE
29     021772 052765 000004 000014 BIS #CT.CMD,C.FLG(R5)       :MARK COMMAND ISSUED
30     022000          POP <R1,R0>
31     022002 012601          MOV (SP)+,R1
32     022004 012600          MOV (SP)+,R0
33     022004 000207          RETURN
    
```

```

1      :CLRBUF
2      :
3      :CLEAR THE SPECIFIED DATA BUFFER IN THE HOST COMM AREA
4      :AND LOAD BUFFER DESCRIPTOR IN COMMAND PACKET TO THE BUFFER
5      :
6      :INPUTS:
7      :   R5 - CONTROLLER TABLE ADDRESS
8      :   R4 - ADDRESS OF HOST COMM AREA
9      :   R0 - OFFSET INTO HOST COMM AREA TO DATA BUFFER
10     :
11     :OUTPUTS:
12     :   DATA BUFFER CLEARED
13     :   COMMAND PACKET POINTING TO BUFFER
14     :   BYTE COUNT SET TO SIZE OF BUFFER
15     :   R4 - ADDRESS OF DATA BUFFER
16     CLRBUF: PUSH <R0,R1>
17     022006 010046          MOV R0,-(SP)
18     022010 010146          MOV R1,-(SP)
19     022012 060400          ADD R4,R0          ;ADD START OF HOST COMM AREA TO OFFSET
20     022014 010064 000124  MOV R0,HC.CPK+P.UADR(R4) ;PUT BUFFER ADDRESS IN COMMAND PACKET
21     022020 012764 000074 000120 MOV #HC.BSZ,HC.CPK+P.BCNT(R4) ;PUT SIZE OF BUFFER IN COMMAND PACKET
22     022026 010004          MOV R0,R4          ;PUT BUFFER ADDRESS IN R4
23     022030 012701 000036  MOV #HC.BSZ/2,R1      ;GET SIZE OF BUFFER IN WORDS
24     022034 005020          CLRBF: CLR (R0)+    ;CLEAR ALL THE WORDS
25     022036 005301          DEC R1
26     022040 001375          BNE CLRBF
27     022042 012601          POP <R1,R0>
28     022044 012600          MOV (SP)+,R1
29     022046 000207          MOV (SP)+,R0
30     RETURN
    
```

```

1      :WAITMS
2
3      :WAIT FOR UDA TO RESPOND WITH A MESSAGE PACKET
4
5      :INPUTS:
6      R5 - ADDRESS OF CONTROLLER TABLE
7      :OUTPUTS:
8      Z CLEAR IF NO ERROR
9      Z SET IF ERROR, MESSAGE PRINTED
10
11     022050      WAITMS: PUSH <R0,R1>
12     022050      010046      MOV R0,-(SP)
13     022052      010146      MOV R1,-(SP)
14     022054      012700      000036      MOV #30,R0      ;SET TIME OUT VALUE OF 30 SECONDS
15     022060      010501      MOV R5,R1      ;POINT TO TIME OUT COUNTER
16     022062      062701      000040      ADD #C.TO,R1
17     022066      004737      022660      CALL SETTO
18     022072      011500      MOV (R5),R0      ;GET ADDRESS OF UDAIP REGISTER
19     022074      032765      000010      000014      1$: BIT #CT.MSG,C.FLG(R5)      ;LOOK IF INTERRUPT OCCURRED
20     022102      001030      BNE 3$      ;BRANCH IF SO
21     022104      016001      000002      MOV 2(R0),R1      ;LOOK AT UDASA REGISTER
22     022110      001034      BNE 4$      ;BRANCH IF ERROR CODE PRESENT
28     022112      104422      BREAK
29     022114      005737      002352      TST KW.CSR      TRAP CSBRK
30     022120      001765      BEQ 1$      ;SEE IF A CLOCK ON SYSTEM
31     022122      023765      002364      000042      CMP KW.EL+2,C.TOH(R5)      ;CHECK IF TIMEOUT HAS HAPPENED
32     022130      101005      BHI 2$
33     022132      001360      BNE 1$
34     022134      023765      002362      000040      CMP KW.EL,C.TO(R5)
35     022142      103754      BLO 1$
36     022144      104455      2$: ERRDF 39,0,RSPTOE      TRAP CSERDF
37     022144      000047      .WORD 39
38     022150      000000      .WORD 0
39     022152      012636      .WORD RSPTOE
40     022154      012601      POP <R1,R0>
41     022156      012600      MOV (SP)+,R1
42     022160      000264      MOV (SP)+,R0
43     022162      000207      SEZ
44     022164      042765      000010      000014      3$: BIC #CT.MSG,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
45     022172      012601      POP <R1,R0>
46     022174      012600      MOV (SP)+,R1
47     022176      000244      MOV (SP)+,R0
48     022200      000207      CLZ
49     022202      000207      RETURN      ;GIVE NO ERROR RETURN
50     022202      104455      4$: ERRDF 40,0,RSPTME      TRAP CSERDF
51     022204      000050      .WORD 40
52     022206      000000      .WORD 0
53     022210      012662      .WORD RSPTME
54     022212      012601      POP <R1,R0>
55     022214      012600      MOV (SP)+,R1
56     022216      000264      MOV (SP)+,R0
57     SEZ
    
```

47 022220 000207

RETURN

1
2
3
4
5
6
7
8
9
10
11
12
13
14 022222 016001 000002
15 022226 006301
16 022230 011002
17 022232 100001
18 022234 005201
19 022236 000207

:APRINT
:
:CONVERT AN 18 BIT ADDRESS STORED IN TWO WORDS INTO A FORMAT
:THAT WILL ALLOW PRINTING OF THE 18 BIT NUMBER.
:
:INPUTS:
: R0 - ADDRESS OF TWO WORD BLOCK CONTAINING ADDRESS.
: FIRST WORD CONTAINING LOW 16 BITS.
: SECOND WORD CONTAINING HIGH 2 BITS.
:
:OUTPUTS:
: R1 - HIGH 3 BITS OF ADDRESS
: R2 - LOW 15 BITS OF ADDRESS
:
APRINT: MOV 2(R0),R1 :GET HIGH 2 BITS
ASL R1 :SHIFT LEFT
MOV (R0),R2 :GET LOW 16 BITS
BPL APRIZ :IF 16TH BIT SET
INC R1 :PLACE IT IN WITH HIGH 2 BITS
APRIZ: RETURN

```

1      :INITKT
2
3      :INITIALIZE THE KT OPTION (IF AVAILABLE) SO THAT IT CAN BE TURNED
4      :ON AND OFF BY KTREAD AND KTWRITE ROUTINES.
5
6      :INPUTS:
7      :      NONE
8
9      :OUTPUTS:
10     :      ALL KT REGISTERS LOADED SO THAT IF THE KT OPTION IS TURNED ON, ALL
11     :      MEMORY WOULD APPEAR THE SAME AS WHEN TURNED OFF. KT OPTION IS TURNED
12     :      OFF IF ON WHEN CALLED.
13
14     :      KTMEM - 0 IF KT EXISTS, -1 IF KT DOES NOT EXIST.
15
16     INITKT: PUSH R0
17           MOV R0,-(SP)
18           CLR NXMAD           ;CLEAR MEMORY FLAG
19           SETVEC #4,#NXMI,#PRI07 ;TRAP NON-EXISTANT ADDRESSES
20
21           MOV #PRI07,-(SP)
22           MOV #NXMI,-(SP)
23           MOV #4,-(SP)
24           MOV #3,-(SP)
25           TRAP C$SVEC
26           ADD #10,SP
27
28           CLR SRO           ;MAKE SURE MEMORY MANAGEMENT IS OFF
29           MOV #MM.PLF+MM.ACF,R0 ;FILL PDR'S
30           MOV R0,PDR0       ;TO ALLOW READ AND WRITE
31           MOV R0,PDR1       ;TO ALL LOCATIONS
32           MOV R0,PDR2
33           MOV R0,PDR3
34           MOV R0,PDR4
35           MOV R0,PDR5
36           MOV R0,PDR6
37           MOV R0,PDR7
38           CLR PAR0           ;POINT PAGE ADDRESS REGISTERS
39           MOV #200,PAR1      ;AT SAME PHYSICAL ADDRESS
40           MOV #400,PAR2
41           MOV #600,PAR3
42           MOV #1000,PAR4
43           MOV #1200,PAR5
44           MOV #1400,PAR6
45           MOV #7600,PAR7
46
47           CLRVEC #4         ;RELEASE THE VECTOR
48
49           MOV #4,R0         MOV #4,R0
50           TRAP C$CVEC       TRAP C$CVEC
51
52           MOV NXMAD,KTMEM    ;STORE KT EXISTS FLAG
53           CLR KTBASO        ;CLEAR OFFSET TO REACH BASE
54           CLR KTBASA        ;CLEAR BASE ADDRESS
55           POP R0
56           MOV (SP)+,R0
57
58           RETURN
    
```



```
1      ;KTBASE
2      ;
3      ;SET BASE ADDRESS FOR BUFFER IN MEMORY. KT READ AND KTWRITE CALLS
4      ;WILL THEN REFERENCE THE BUFFER USING AN OFFSET THAT GETS ADDRESS
5      ;RELATIVE TO BASE.
6      ;
7      ;INPUTS:
8      ;      R0 - ADDRESS OF TWO WORD BLOCK CONTAINING 18 BIT ADDRESS
9      ;
10     ;OUTPUTS:
11     ;      NONE
12     KTBASE: PUSH R0
13     ;      MOV R0,-(SP)
14     ;      MOV B (R0)+,KTBAS0      ;FIRST BYTE IS OFFSET
15     ;      MOV B (R0)+,KTBAS1      ;SECOND AND THIRD BYTES
16     ;      MOV B (R0)+,KTBAS2      ; MAKE THE BASE ADDRESS
17     ;      POP R0
18     ;      MOV (SP)+,R0
19     ;      RETURN
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

022474
022474 010146
022476 063701 002222
022502
022502 010146
022504 000301
022506 042701 177400
022512 063701 002220
022516 006301
022520 006301
022522 010137 172354
022526
022526 012601
022530 042701 177400
022534 012737 000001 177572
022542 016100 140000
022546 005037 177572
022552
022552 012601
022554 000207

```

:KTREAD
:READ ONE WORD FROM MEMORY USING KT HARDWARE. WORD IS SPECIFIED BY A BASE
:ADDRESS AND A BYTE OFFSET. THE BASE ADDRESS MUST BE PREVIOUSLY SPECIFIED
:BY A KTBASE CALL.
:INPUTS:
:   R1 - BYTE OFFSET INTO BUFFER
:OUTPUTS:
:   R0 - DATA READ FROM MEMORY LOCATION

KTREAD: PUSH R1
        MOV R1,-(SP)
        ADD KTBASO,R1           ;COMPUTE TOTAL OFFSET
        PUSH R1
        MOV R1,-(SP)
        SWAB R1                ;GET RID OF LOW BYTE
        BIC #177400,R1
        ADD KTBASA,R1           ;ADD BASE ADDRESS
        ASL R1                  ;SHIFT TO POSITIN FOR PAR REGISTER
        ASL R1
        MOV R1,PAR6            ;STORE IN PAR REGISTER
        POP R1
        MOV (SP)+,R1
        BIC #177400,R1         ;CLEAR HIGH BYTE
        MOV #MM.EN,SRO        ;TURN ON MEMORY MANAGEMENT
        MOV 140000(R1),R0     ;READ WORD
        CLR SRO               ;TURN OFF MEMORY MANAGEMENT
        POP R1
        MOV (SP)+,R1
        RETURN
    
```

```

1      :KTWRITE
2      :
3      :WRITE TO ONE WORD OF MEMORY USING KT HARDWARE. WORD IS SPECIFIED
4      :BY A BASE ADDRESS AND A BYTE OFFSET. THE BASE ADDRESS MUST BE
5      :PREVIOUSLY SPECIFIED BY A KTBASE CALL.
6      :
7      :INPUTS:
8      :       R0 - DATA TO BE WRITTEN
9      :       R1 - BYTE OFFSET INTO BUFFER
10     :
11     :OUTPUTS:
12     :       NONE
13     :
14     :KTWRITE:PUSH R1
15     :       MOV R1,-(SP)
16     :       ADD KTBASO,R1           ;COMPUTE TOTAL OFFSET
17     :       PUSH R1
18     :       MOV R1,-(SP)
19     :       SWAB R1                 ;GET RID OF LOW BYTE
20     :       BIC #177400,R1
21     :       ADD KTBASA,R1         ;ADD BASE ADDRESS
22     :       ASL R1                 ;SHIFT TO POSITION FOR PAR REGISTER
23     :       ASL R1
24     :       MOV R1,PAR6          ;STORE IN PAR REGISTER
25     :       POP R1
26     :       MOV (SP)+,R1
27     :       BIC #177400,R1       ;CLEAR HIGH BYTE
28     :       MOV #MM.EN,SRO      ;TURN ON MEMORY MANAGEMENT
29     :       MOV R0,140000(R1)   ;WRITE WORD
30     :       CLR SRO             ;TURN OFF MEMORY MANAGEMENT
31     :       POP R1
32     :       MOV (SP)+,R1
33     :       RETURN

```

13	022556			
	022556	010146		
14	022560	063701	002222	
15	022564			
	022564	010146		
16	022566	000301		
17	022570	042701	177400	
18	022574	063701	002220	
19	022600	006301		
20	022602	006301		
21	022604	010137	172354	
22	022610			
	022610	012601		
23	022612	042701	177400	
24	022616	012737	000001	177572
25	022624	010061	140000	
26	022630	005037	177572	
27	022634			
	022634	012601		
28	022636	000207		

```
1      :NXMI
2      :
3      :NON-EXISTANT MEMORY SERVICE ROUTINE
4      :
5      :INPUTS:
6      :      NXMAD SET TO ZERO
7      :
8      :OUTPUTS:
9      :      NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
10     022640      BGNSRV NXMI
11     022640
12     022640 012737 177777 002366      MOV #-1,NXMAD
13
14     022646      ENDSRV
15     022646
16     022646 000002

NXMI::
L10021:
RTI
```

```
1      :UDASRV
2      :
3      :UDA INTERRUPT SERVICE ROUTINE. MARKS UDA CONTROLLER TABLE THAT AN
4      :INTERRUPT HAS BEEN RECEIVED.
5      :
6      :THIS ROUTINE IS CALLED BY A [JSR R0,UDASRV] INSTRUCTION FROM WITHIN
7      :THE CONTROLLER TABLE. THE PC STORED IN R0 IS THE ADDRESS OF THE C.FLG
8      :WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
9      :OF R0 FOLLOWED BY THE INTERRUPTED PC AND PS.
10     :
11     :INPUTS:
12     :      R0 - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
13     :      STACK - SAVED CONTENTS OF R0
14     :OUTPUTS:
15     :      CT.CMD CLEARED AND CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
16     :      R0 - RESTORED FROM STACK
17     :
18 022650 BGNSRV UDASRV
19 022650
20 022650 052710 000010      ;      BIC #CT.CMD,(R0)          ;CLEAR CT.CMD
21 022654      ;      BIS #CT.MSG,(R0)        ;SET CT.MSG
22 022654 012600      ;      POP R0                ;RESTORE R0
23 022656      ;      MOV (SP)+,R0
24 022656      ENDSRV
25 022656 000002      L10022:      RTI
```

1				:SETTO	
2				:SET TIMEOUT COUNTER TO SOME NUMBER OF SECONDS FROM CURRENT TIME.	
3				:INPUTS:	
4				R0 - NUMBER OF SECONDS FOR TIMEOUT	
5				R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT	
6				:OUTPUTS:	
7				R0 - CONTENTS DESTROYED	
8				R1 - INCREMENTED BY 2	
9					
10				:COMPUTE CLOCK TICKS TIL TIMEOUT	
11				SETTO: PUSH <R2,R3>	
12				MOV R2,-(SP)	
13				MOV R3,-(SP)	
14	022660			CLR R2	:CLEAR PRODUCT
15	022660	010246		MOV KW.HZ,R3	:GET MULTIPLICAND
16	022662	010346		SET00: ASR R0	:SHIFT MULTIPLIER TO RIGHT
17	022664	005002	002360	BCC SET01	:IF A ONE BIT SHIFTED OUT
18	022666	013703		ADD R3,R2	: ADD MULTIPLICAND TO PRODUCT
19	022672	006200		SET01: ASL R3	:DOUBLE THE MULTIPLICAND
20	022674	103001		TST R0	
21	022676	060302		BNE SET00	:CONTINUE UNTIL MULTIPLIER IS ZERO
22	022700	006303			
23	022702	005700			
24	022704	001372			
25				:GET CURRENT TIME	
26	022706	013700	002362	SET02: MOV KW.EL,R0	:GET TIME
27	022712	013703	002364	MOV KW.EL+2,R3	
28	022716	020037	002362	CMP R0,KW.EL	:IF CHANGED DURING RETRIEVAL
29	022722	001371		BNE SET02	: GET IT AGAIN
30					
31				:ADD TIME TIL TIMEOUT	
32				ADD R2,R0	:ADD
33	022724	060200		ADC R3	
34	022726	005503			
35					
36				:PUT RESULT IN STORAGE	
37				MOV R0,(R1)+	
38	022730	010021		MOV R3,(R1)	
39	022732	010311			
40					
41	022734			POP <R3,R2>	
42	022734	012603		MOV (SP)+,R3	
	022736	012602		MOV (SP)+,R2	
	022740	000207		RETURN	

```

1      :UDAIN
2
3      :FUNCTIONAL DESCRIPTION:
4          SUBROUTINE TO INITIALIZE A UDA AND BRING IT ON-LINE.
5          ALL STEPS ARE CHECKED. AN ERROR MESSAGE IS REPORTED IF ANY ERROR
6          DETECTED.
7
8      :INPUTS:
9          R5 - ADDRESS OF CONTROLLER TABLE.
10         R4 - LEN, INTI AND VECTOR FIELDS TO SEND TO UDA
11      :IMPLICIT INPUTS:
12         FFREE - FIRST FREE ADDRESS OF MEMORY. THIS ADDRESS IS GIVEN TO UDA
13         AS START OF RING BUFFER.
14         FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS.
15      :OUTPUTS:
16         CONDITION Z - SET IF ANY ERROR REPORTED. CLEAR IF NO ERROR.
17         R1 - SIZE OF RING BUFFER IN WORDS IF NO ERROR.
18         R4 - ADDRESS OF UDAIP REGISTER IN UDA
19         R5 - UNCHANGED.
20
21      :CHECK IF ENOUGH FREE MEMORY FOR RING BUFFER
22
23 022742 010400      UDAIN: MOV R4,R0          ;GET MESSAGE LENGTH
24 022744 000300      SWAB R0
25 022746 042700 177770 BIC #177770,R0
26 022752 004737 023654 CALL CLOG          ;COMPUTE LOGARITHMIC VALUE
27 022756 010102      MOV R1,R2          ;SAVE RESULT IN R2
28 022760 010400      MOV R4,R0          ;GET COMMAND LENGTH
29 022762 000300      SWAB R0
30 022764 006000      ROR R0
31 022766 006000      ROR R0
32 022770 006000      ROR R0
33 022772 042700 177770 BIC #177770,R0
34 022776 004737 023654 CALL CLOG          ;COMPUTE LOGARITHMIC VALUE
35 023002 060201      ADD R2,R1          ;ADD THE TWO RESULTS
36 023004 006301      ASL R1             ;MULTIPLY BY 2 WORDS PER RING
37 023006 062701 000002 ADD #HC.ISZ/2,R1   ;ADD SPACE FOR INTERRUPT INDICATORS
38 023012 020137 002176 CMP R1,FSIZE       ;COMPARE WITH SIZE OF FREE MEMORY
39 023016 101402      BLOS UDAI1
40 023020 000137 012710 JMP FMERR          ;FATAL ERROR IF NOT ENOUGH MEMORY
    
```

```

1          :FILL HOST COMMUNICATION AREA WITH ALL ONES
2
3 023024 013702 002174   UDAI1:  MOV FFREE,R2           :GET FIRST ADDRESS OF RING BUFFER
4 023030 010103         MOV R1,R3           :GET SIZE OF RING BUFFER
5 023032 012722 177777   UDAI1L: MOV #-1,(R2)+       :WRITE ONES TO BUFFER
6 023036 005303         DEC R3           :COUNT THE WORDS IN BUFFER
7 023040 003374         BGT UDAI1L        :LOOP UNTIL ENTIRE BUFFER WRITTEN
8
9          :DO THE INITIALIZATION
10
11 023042 004737 023204   CALL UDAIST           :DO FIRST THREE STEPS
12 023046 103454         BCS UDAIEX        :GET OUT IF UDA MICROCODE REPORTED FAILURE
13 023050 012364 000002   MOV (R3)+,2(R4)       :WRITE NEXT WORD TO UDASA REGISTER
14 023054 012703 000310   MOV #200,R3          :GET TRY COUNTER
15 023060 016402 000002   UDAI1A: MOV 2(R4),R2    :LOOK AT UDASA
16 023064 001410         BEQ UDAI1C
17 023066 100005         BPL UDAI1B
18 023070         ERRDF 7,INTHD,INTR5
19 023070 104455         TRAP C$ERDF
20 023072 000007         .WORD 7
21 023074 003263         .WORD INTHD
22 023076 012200         .WORD INTR5
23 023100 000437         BR UDAIEX
24 023102 005303   UDAI1B: DEC R3
25 023104 001365         BNE UDAI1A
26 023106 010264 000002   UDAI1C: MOV R2,2(R4)       :WRITE 0 TO UDASA (PURGE)
27 023112 011402         MOV (R4),R2     :READ FROM UDAIP (POLL)
28 023114 004737 023514   CALL UDARSP          :WAIT FOR STEP OR ERROR BIT
29 023120 103427         BCS UDAIEX        :GET OUT IF UDA MICROCODE REPORTED FAILURE
30
31          :CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
32
33 023122 013702 002174   UDAI2:  MOV FFREE,R2           :GET FIRST ADDRESS OF RING BUFFER
34 023126 010103         MOV R1,R3           :GET SIZE OF RING BUFFER
35 023130 005722   UDAI2L: TST (R2)+       :CHECK WORD IN BUFFER
36 023132 001003         BNE UDAI2E        :GO TO ERROR REPORTER IF NOT ZERO
37 023134 005303         DEC R3           :COUNT THE WORDS IN BUFFER
38 023136 003374         BGT UDAI2L        :LOOP UNTIL ALL WORDS CHECKED
39 023140 000405         BR UDAI3
40
41 023142         UDAI2E: ERRDF 15,INTHD,INTBF   :REPORT BUFFER NOT CLEARED
42 023142 104455         TRAP C$ERDF
43 023144 000017         .WORD 15
44 023146 003263         .WORD INTHD
45 023150 012244         .WORD INTBF
46 023152 000412         BR UDAIEX
47
48          :SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION
49
50 023154 016500 000006   UDAI3:  MOV C,BST(R5),R0       :GET BURST VALUE
51 023160 006300         ASL R0           :SHIFT TO POSITION
52 023162 006300         ASL R0
53 023164 052700 000001   BIS #SA.GO,R0        :SET THE GO BIT
54 023170 010064 000002   MOV R0,2(R4)         :SEND TO UDA
55 023174 000244         CLZ           :CLEAR Z AS NO ERROR INDICATION
56 023176 000207         RETURN

```


50
51
52 023200 000264
53 023202 000207

;ERROR RETURN

UDAIEX: SEZ
RETURN

;SET Z TO INDICATE ERROR OCCURRED

```

1      :UDAIST
2
3      :START THE INITIALIZATION PROCESS ON THE SELECTED UDA.
4      :STOP BEFORE WRITING THE THIRD WORD SO UDA DOES NOT
5      :ATTEMPT ANY UNIBUS TRANSFERS.
6
7      :INPUTS:
8      :       R5 - ADDRESS OF CONTROLLER TABLE
9      :       R4 - LEN, INTI AND VECTOR FIELDS TO SEND TO UDA
10
11     :LOAD TABLE OF DATA TO SEND TO UDASA REGISTER
12
13     UDAIST: BREAK
14     023204      104422      TRAP      C$BRK
15     023206      010146      PUSH R1
16     023210      052704      100000      MOV R1,-(SP)
17     023214      010437      023406      BIS #SA.STP,R4      ;SET STEP BIT IN DATA WORD
18     023220      013737      002174      023412      MOV R4,UDAID1      ;LOAD LENGTH AND INTERRUPT VECTOR
19     023226      062737      000004      023412      MOV FFREE,UDAID2   ;LOAD MEMORY ADDRESS
20     :START THE INITIALIZATION BY WRITING TO UDAIP REGISTER
21
22     023234      016504      000000      MOV C.UADR(R5),R4   ;GET ADDRESS OF UDAIP REGISTER
23     023240      005037      002366      CLR NXMAD           ;CLEAR MEMORY ERROR FLAG
24     023244      012746      000340      SETVEC #4,#NXMI,#PRI07 ;SET UP VECTOR 4
25     023244      012746      022640      MOV #PRI07,-(SP)
26     023250      012746      000004      MOV #NXMI,-(SP)
27     023254      012746      000004      MOV #4,-(SP)
28     023260      012746      000003      MOV #3,-(SP)
29     023264      104437      TRAP      C$SVEC
30     023266      062706      000010      ADD #10,SP
31     023272      005764      000002      TST 2(R4)           ;ACCESS UDASA REGISTER
32     023276      005014      CLR (R4)           ;WRITE TO UDAIP
33     023300      012700      000004      CLRVEC #4           ;GIVE UP THE VECTOR
34     023304      104436      MOV #4,R0
35     023306      005737      002366      TRAP C$CVEC
36     023312      001406      TST NXMAD           ;SEE IF A MEMORY ERROR OCCURRED
37     023314      104455      BEQ UDAISG
38     023316      000025      ERRDF 21,0,INTR4
39     023320      000000      TRAP      C$ERDF
40     023322      012154      .WORD 21
41     023324      000261      .WORD 0
42     023326      000424      .WORD INTR4
43
44     SEC
45     BR UDAISE
46
47     :SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
48
49     023330      012737      004000      023652      UDAISG: MOV #SA.S1,UDARSD ;STORE RESPONSE MASK
50     023336      012703      023404      MOV #UDAIDT,R3     ;AND INDEX TO TABLE
51
52     :WAIT FOR AND CHECK RESPONSE DATA
53
54     023342      004737      023514      UDAISL: CALL UDARSP      ;WAIT FOR STEP OR ERROR BITS
55     023346      103414      BCS UDAISE        ;EXIT IF ERROR
56     023350      004733      CALL @(R3)+       ;CALL RESPONSE CHECKER FOR STEP
    
```

```
44 023352 103412          BCS UDAISE          ;GET OUT IF ERROR
45 023354 006337 023652  ASL UDARSD         ;SHIFT TO NEXT STEP BIT
46 023360 032737 040000 023652 BIT #SA,S4,UDARSD  ;CHECK IF NOW AT STEP 4
47 023366 001003          BNE UDAISX         ;GET OUT IF SO
48 023370 012364 000002  MOV (R3)+,2(R4)    ;WRITE DATA TO UDASA REGISTER
49 023374 000762          BR UDAISL          ;STAY IN LOOP
50
51 023376 000241          UDAISX: CLC          ;CLEAR CARRY FOR NO ERROR INDICATION
52 023400          UDAISE: POP R1
    023400 012601          MOV (SP)+,R1
53 023402 000207          RETURN
```

```

1          ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION
2
3 023404 023422  UDAIDT: .WORD UDAIR1          ;FIRST WORD RESPONSE CHECK ROUTINE
4 023406 000000  UDAID1: .WORD 0          ;FIRST WORD TO SEND TO UDASA
5 023410 023430          .WORD UDAIR2          ;SECOND WORD RESPONSE CHECK ROUTINE
6 023412 000000  UDAID2: .WORD 0          ;SECOND WORD TO SEND TO UDASA
7 023414 023450          .WORD UDAIR3          ;THIRD WORD RESPONSE CHECK ROUTINE
8 023416 100000  UDAID3: .WORD SA.TST        ;THIRD WORD TO SEND TO UDASA
9 023420 023466          .WORD UDAIR4          ;FOURTH WORD RESPONSE CHECK ROUTINE
10
11         ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
12         ;CHECK FOR PROPER CONTROLLER TYPE
13
14 023422 012701 004400  UDAIR1: MOV #SA.S1+SA.DI,R1      ;SET STEP ONE BIT
15 023426 000422          BR UDAIRC          ;NOW COMPARE
16
17         ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
18         ;CHECK FOR ECHO OF INTI AND VECTOR
19
20 023430 013701 023406  UDAIR2: MOV UDAID1,R1          ;GET WORD SENT TO UDASA
21 023434 000301          SWAB R1          ;GET HIGH 8 BITS
22 023436 042701 177400          BIC #177400,R1
23 023442 052701 010000          BIS #SA.S2,R1      ;SET STEP 2 BIT
24 023446 000412          BR UDAIRC          ;NOW COMPARE
25
26         ;RESPONSE CHECK FOR THIRD WORD FROM UDASA
27         ;CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
28
29 023450 013701 023406  UDAIR3: MOV UDAID1,R1          ;GET WORD SENT TO UDASA
30 023454 042701 177400          BIC #177400,R1      ;JUST LOW 8 BITS
31 023460 052701 020000          BIS #SA.S3,R1      ;SET STEP 3 BIT
32 023464 000403          BR UDAIRC          ;NOW COMPARE
33
34         ;RESPONSE CHECK FOR FOURTH WORD FROM UDASA
35         ;CHECK FOR ECHO OF PURGE AND LFAIL BITS
36
37 023466 010201          UDAIR4: MOV R2,R1          ;GET RESPONSE FROM UDA
38 023470 042701 137400          BIC #^C<SA.S4+SA.MCV>,R1 ;KEEP MICROCODE VERSION AND STEP 4
39
40         ;COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
41
42 023474 020102  UDAIRC: CMP R1,R2          ;COMPARE THE DATA
43 023476 001405          BEQ UDAIRX         ;EXIT IF COMPARED CORRECTLY
44 023500          ERRDF 6,INTHD,INTR3      ;REPORT ERROR
45 023500 104455          TRAP C$ERDF
46 023502 000006          .WORD 6
47 023504 003263          .WORD INTHD
48 023506 012066          .WORD INTR3
49 023510 000261
50 023512 000207  UDAIRX: RETURN
    
```

```

1      :UDARSP
2
3      :WAIT FOR UDA TO RESPOND WITH DATA IN UDASA REGISTER.
4      :EITHER STEP BIT FROM MASK IN LOCATION UDARSD OR ERROR BIT
5      :WILL CAUSE A TERMINATION.
6      :AN ERROR MESSAGE WILL BE PRINTED IF THE UDA DOES NOT RESPOND
7      :IN 10 SECONDS OR IF ERROR SETS.
8
9      :INPUTS:
10     :UDASRD - MASK OF STEP BIT TO LOOK FOR
11     :R5 - ADDRESS OF CONTROLLER TABLE
12     :R4 - ADDRESS OF UDAIP REGISTER
13     :OUTPUTS:
14     :ERROR MESSAGE IF TIME OUT ON RESPONSE OR ERROR BIT SETS
15     :R2 - DATA FROM UDASA REGISTER
16     :CARRY SET IF ERROR BIT SETS OR TIME OUT
17
18 023514 UDARSP: PUSH R1
19 023514 010146          MOV R1,-(SP)
20 023516 052737 100000 023652  BIS #SA.ERR,UDARSD      ;SET ERROR BIT IN MASK WORD
21 023524 012700 000012      MOV #10,,R0          ;SET UP FOR 10 SECOND TIMEOUT
22 023530 010501          MOV R5,R1            ;POINT TO COUNTER IN CONTROLLER TABLE
23 023532 062701 000040      ADD #C.TO,R1
24 023536 004737 022660      CALL SETTO
25 023542 012601          POP R1
26 023544 033764 023652 000002 UDARS1: MOV (SP)+,R1
27 023552 001024          BIT UDARSD,2(R4)      ;LOOK AT ERROR AND STEP BIT
28 023554 104422          BNE UDARS2          ;BRANCH IF EITHER SET
29 023556 005737 002352      BREAK
30 023562 001770          TST KW.CSR          TRAP    C$BRK
31 023564 023765 002364 000042 BEQ UDARS1          ;SEE IF CLOCK ON SYSTEM
32 023572 101005          CMP KW.EL+2,C.TOH(R5) ;CHECK IF TIME OUT OCCURRED
33 023574 001363          BHI 1$
34 023576 023765 002362 000040 BNE UDARS1
35 023604 103757          CMP KW.EL,C.TO(R5)
36 023606 016402 000002 1$: BLO UDARS1
37 023612 000004          MOV 2(R4),R2      ;GET REGISTER CONTENTS
38 023614 104455          ERRDF 4,INTHD,INTR1 ;REPORT TIME OUT ERROR
39 023616 000004          TRAP    C$ERDF
40 023620 003263          .WORD 4
41 023622 011756          .WORD INTHD
42 023622 000407          .WORD INTR1
43
44 BR UDARSE
    
```

```

1          ;CHECK IF ERROR BIT SET
2
3 023624 016402 000002 UDARS2: MOV 2(R4),R2          ;GET REGISTER CONTENTS
4 023630 100006          BPL UDARSX          ;EXIT IF ERROR NOT SET
5 023632          ERRDF 5,INTHD,INTR2        ;REPORT ERROR INFO
   023632 104455          TRAP C$ERDF
   023634 000005          .WORD 5
   023636 003253          .WORD INTHD
   023640 012022          .WORD INTR2
6 023642 000261 UDARSE: SEC
7 023644 000207          RETURN
8
9          ;NORMAL EXIT
10
11 023646 000241 UDARSX: CLC          ;CLEAR CARRY AS NO ERROR INDICATION
12 023650 000207          RETURN
13
14          ;LOCATION FOR STEP BIT MASK
15
16 023652 000000 UDARSD: .WORD 0          ;LOAD BY CALLING ROUTINE

```

1
2
3
4
5
6
7
8
9

10 023654
023654 010046
11 023656 005001
12 023660 000261
13 023662 006101
14 023664 005300
15 023666 100375
16 023670
023670 012600
17 023672 000207

```
:CLOG  
:COMPUTE LOGARITHMIC VALUE OF NUMBER TO BASE 2.  
:INPUTS:  
:   R0 - LOGARITHM TO BE CONVERTED  
:OUTPUTS:  
:   R1 - VALUE OF 2 RAISED TO POWER OF INPUT NUMBER  
CLOG:  PUSH R0  
      MOV R0,-(SP)  
      CLR R1  
      SEC  
CLOGLP: ROL R1  
      DEC R0  
      BPL CLOGLP  
      POP R0  
      MOV (SP)+,R0  
      RETURN  
      ;SET UP ZERO START VALUE  
      ;WITH CARRY READY TO SHIFT IN  
      ;SHIFT TO LEFT  
      ; UNTIL R0  
      ; GOES NEGATIVE
```

```
1      ;RDDLL
2      ;
3      ;READ DISK DRIVE DOWNLINE LOAD PROGRAM INTO MEMORY
4      ;
5      ;INPUTS:
6      ;      DLLNAM - NAME OF PROGRAM IN RAD50 (TWO WORDS)
7      ;
8      ;OUTPUTS:
9      ;      FREE MEMORY CONTAINING PROGRAM
10     ;      CARRY CLEAR IF NO ERROR, CARRY SET IF PROGRAM NOT FOUND
11     023674 012701 000005      RDDLL:  MOV #5,R1          ;TYPE OF PROGRAM IN DATA FILE
12     023700 004737 023734      CALL RDREC        ;READ PROGRAM INTO MEMORY
13     023704 006101              ROL R1           ;PRESERVE CARRY STATE IN R1
14     023706 004737 023716      CALL CLOSEF      ; WHILE CLOSING THE DATA FILE
15     023712 006001              ROR R1           ; AS NORMAL POSITION IS LOST
16     023714 000207              RETURN
```



```
1      :CLOSEF
2
3      :CLOSE DATA FILE FOR DM PROGRAMS
4
5      :INPUTS:
6          FILOPN - ZERO IF FILE NOT OPEN
7      :OUTPUTS:
8          NONE
9
10     023716 005737 002256    CLOSEF: TST FILOPN          ;SEE IF FILE CURRENTLY OPEN
11     023722 001403          BEQ 1$
12     023724          CLOSE          ; IF SO, CLOSE IT
13     023724 104435          ;AND MARK AS SO          TRAP    C$CLOS
14     023726 005037 002256
15     023732 000207          1$: CLR FILOPN
16                                     RETURN
```

```

1      :RDREC
2
3      :READ A RECORD FROM THE INPUT FILE. PLACE DATA INTO FREE MEMORY.
4
5      :INPUTS:
6      :   R1 - FILE TYPE
7          1 - TEST 1 DM PROGRAM
8          2 - TEST 2 DM PROGRAM
9          3 - TEST 3 DM PROGRAM
10         4 - TEST 4 DM PROGRAM
11         5 - DRIVE DIAGNOSTIC DOWNLINE LOAD PROGRAM
12         DLLNAM - IF R1 CONTAINS 5, TWO WORDS AT THIS ADDRESS CONTAIN
13         NAME OF PROGRAM IN RAD50.
14      :OUTPUTS:
15         DATA FROM RECORD IN MEMORY
16         CARRY CLEAR IF NO ERROR, CARRY SET IF ERROR
17
18      RDREC: PUSH <R0,R1,R2,R3,R4,R5>
19      023734 010046          MOV R0,-(SP)
20      023736 010146          MOV R1,-(SP)
21      023740 010246          MOV R2,-(SP)
22      023742 010346          MOV R3,-(SP)
23      023744 010446          MOV R4,-(SP)
24      023746 010546          MOV R5,-(SP)
25
26      19 023750 005737 002256  TST FILOPN          ;SEE IF FILE ALREADY OPEN
27      20 023754 001006          BNE RDST
28      21 023756          OPEN #FNAME          ;IF NOT, OPEN FILE NOW
29      22 023756 012700 002240          MOV #FNAME,R0
30      23 023762 104434          TRAP C$OPEN
31      24 023764 005237 002256          INC FILOPN          ;AND MARK AS OPEN
32      25 023770 005005          CLR R5             ;CLEAR LOAD ADDRESS (SEARCH MODE)
33      26 023772          BREAK             ;ALLOW PROGRAM TO BE INTERRUPTED
34      27 023774          GETBYTE R4          TRAP C$BRK
35      28 023774 104426          ;GETBYTE CALLS DON'T SEEM TO BREAK ON CONTROL-C!
36      29 023776 110004          ;GET A BYTE
37      30 024000 005704          TRAP C$GETB
38      31 024002 001773          MOVB RO,R4
39      32 024004 022704 000001          TST R4             ;IF ZERO
40      33 024010 001153          BEQ RDST           ;KEEP READING
41      34 024012 104426          CMP #1,R4         ;WHEN NOT ZERO
42      35 024014 060004          BNE RWRDE1        ;IT BETTER BE A ONE
43      36 024016 005700          GETBYTE R0        ;AND THE NEXT BYTE
44      37 024020 001432          TRAP C$GETB
45      38 024022 020001          ADD R0,R4
46      39 024024 103430          TST R0             ;IF ZERO, PROCESS DATA
47      40 024026 101132          BEQ RDDAT
48      41 024030 004737 024272          CMP R0,R1
49      42 024034 013702 002254          BLO RDDAT
50      43 024040 004737 024272          BHI RDERR
51      44 024044          CALL FWORD
52      45 024046 060004          MOV FDATA,R2
53      46 024050 105704          CALL FWORD
54      47 024052 001133          GETBYTE R0
55      48 024054          ;GET CHECKSUM
56      49 024056          TRAP C$GETB
57      50 024058          ADD R0,R4
58      51 024060          ;ADD TO COMPUTED SUM
59      52 024062          TSTB R4
60      53 024064          ;SEE IF THIS SUM IS ZERO
61      54 024066          BNE RWRDE3
62      55 024068          ;IF NOT, REPORT CHECKSUM ERROR
    
```

45	024054	020127	000005		CMP R1,#5		:IF FILE TYPE IS A 5
46	024060	001007			BNE 1\$		
47	024062	023702	002430		CMP DLLNAM,R2		: MATCH THE PROGRAM NAME
48	024066	001341			BNE RDST		:KEEP SEARCHING IF NOT DESIRED PROGRAM
49	024070	023737	002432	002254	CMP DLLNAM+2,FDATA		
50	024076	001335			BNE RDST		
51	024100	013705	002174	1\$:	MOV FFREE,R5		:GET STORAGE ADDRESS
52	024104	000732			BR RDST		:SWITCH FROM SEARCH TO STORE MODE
53							
54	024106	004737	024272	RDDAT:	CALL FWORD		:READ BYTE COUNT
55	024112	013703	002254		MOV FDATA,R3		:SAVE IN R3
56	024116	004737	024272		CALL FWORD		:READ LOAD ADDRESS
57	024122	162703	000006		SUB #6,R3		:SUBTRACT BYTES ALREADY READ FROM BYTE COUNT
58	024126	001443			BEQ RWORDT		:IF RESULT IS ZERO, THIS IS A
59							:TRANSFER BLOCK
60	024130	005705			TST R5		:IF IN SEARCH MODE,
61	024132	001425			BEQ 3\$: BYPASS TRANSFER ADDRESS COMPUTATION
62	024134	013701	002254		MOV FDATA,R1		:GET LOAD ADDRESS
63	024140	060301			ADD R3,R1		:ADD BYTES IN RECORD
64	024142	162701	001000		SUB #DMFRST,R1		
65	024146	060501			ADD R5,R1		:COMPUTE STORAGE ADDRESS
66	024150	032701	000001		BIT #1,R1		:CHECK IF ODD BYTE ADDRESS
67	024154	001401			BEQ 1\$:IF SO,
68	024156	005201			INC R1		: INCREMENT
69	024160	163701	002174	1\$:	SUB FFREE,R1		:SEE IF MORE MEMORY NEEDS TO BE ALLOCATED
70	024164	101403			BLOS 2\$		
71	024166	006001			ROR R1		:REDUCE TO WORDS
72	024170	004737	012722		CALL ALOCM		:ALLOCATE THE MEMORY
73	024174	013702	002254	2\$:	MOV FDATA,R2		:GET LOAD ADDRESS
74	024200	162702	001000		SUB #DMFRST,R2		
75	024204	060502			ADD R5,R2		
76	024206			3\$:	GETBYTE R0		:GET DATA BYTE
	024206	104426					TRAP C\$GETB
77	024210	005705			TST R5		:IF IN SEARCH MODE,
78	024212	001401			BEQ 4\$: BYPASS DATA STORAGE
79	024214	110022			MOVB R0,(R2)+		:STORE IN MEMORY
80	024216	060004		4\$:	ADD R0,R4		:UPDATE CHECKSUM
81	024220	005303			DEC R3		:COUNT THE BYTE
82	024222	001371			BNE 3\$:GET THEM ALL
83	024224				GETBYTE R0		:GET CHECKSUM
	024224	104426					TRAP C\$GETB
84	024226	060004			ADD R0,R4		:ADD
85	024230	105704			TSTB R4		:IF CHECKSUM CORRECT,
86	024232	001657			BEQ RDST		: THEN GO READ NEXT RECORD
87	024234	000446			BR RWRDE3		: ELSE REPORT ERROR
88							
89	024236			RWORDT:	GETBYTE R0		:READ CHECKSUM BYTE
	024236	104426					TRAP C\$GETB
90	024240	060004			ADD R0,R4		:ADD TO COMPUTED CHECKSUM
91	024242	105704			TSTB R4		:CHECK LOW BYTE OF SUM
92	024244	001042			BNE RWRDE3		:BRANCH IF CHECKSUM ERROR
93	024246	005705			TST R5		:IF IN SEARCH MODE,
94	024250	001650			BEQ RDST		: KEEP ON SEARCHING
95	024252				POP <R5,R4,R3,R2,R1,R0>		
	024252	012605			MOV (SP)+,R5		
	024254	012604			MOV (SP)+,R4		
	024256	012603			MOV (SP)+,R3		

```

024260 012602          MOV (SP)+,R2
024262 012601          MOV (SP)+,R1
024264 012600          MOV (SP)+,R0
96 024266 000241      CLC
97 024270 000207      RETURN
98
99 024272          FWORD: GETBYTE R0          ;READ A BYTE FROM FILE
024272 104426          ;UPDATE CHECKSUM ERROR TRAP C$GETB
100 024274 060004      ADD R0,R4          ;START TO BUILD WORD
101 024276 110037 002254  MOVB R0,FDATA      ;READ ANOTHER BYTE FROM FILE
102 024302          GETBYTE R0          TRAP C$GETB
024302 104426          ;UPDATE CHECKSUM
103 024304 060004      ADD R0,R4          ;COMPLETE WORD
104 024306 110037 002255  MOVB R0,FDATA+1
105 024312 000207      RETURN
106
107 024314 004737 023716  RDERR: CALL CLOSEF          ;CLOSE FILE AS POSITION IS LOST
108 024320          POP <R5,R4,R3,R2,R1,R0>
024320 012605          MOV (SP)+,R5
024322 012604          MOV (SP)+,R4
024324 012603          MOV (SP)+,R3
024326 012602          MOV (SP)+,R2
024330 012601          MOV (SP)+,R1
024332 012600          MOV (SP)+,R0
109 024334 000261      SEC
110 024336 000207      RETURN          ;ERROR RETURN, FILE NOT FOUND
    
```

1	024340		RWRDE1: ERRSF 30001,RWRDEM,RWRDM1	;START FRAME NOT FOUND	TRAP	C\$ERSF
	024340	104454			.WORD	30001
	024342	072461			.WORD	RWRDEM
	024344	003310			.WORD	RWRDM1
	024346	012370				
2	024350		DOCLN		TRAP	C\$DCLN
	024350	104444				
3	024352		RWRDE3: ERRSF 30003,RWRDEM,RWRDM3	;CHECKSUM ERROR	TRAP	C\$ERSF
	024352	104454			.WORD	30003
	024354	072463			.WORD	RWRDEM
	024356	003310			.WORD	RWRDM3
	024360	012436				
4	024362		DOCLN		TRAP	C\$DCLN
	024362	104444				

```
1      ;KW11I
2      ;
3      ;CLOCK INTERRUPT SERVICE ROUTINE
4
5 024364      BGNSRV KW11I
6 024364      062737 000001 002362      ADD #1,KW.EL      KW11I::
7 024372      005537 002364      ADC KW.EL+2      ;COUNT THE INTERRUPT
8 024376      012777 000103 155746      MOV #KWOUT.,@KW.CSR      ;RESTART THE CLOCK
9 024404      ENDSRV
   024404
   024404 000002      L10023:
                               RTI
```

1 024406
024406
2 024406 005237 002236
3 024412
024412
024412 000002

BGNSRV INTSRV
INC INTRCV
ENDSRV

; UDA INTERRUPT SERVER
INTSRV::
; FLAG INTERRUPT AS RECEIVED

L10024:
RTI

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10 024414  
    024414 104422  
11 024416  
    024416 104433  
12 024420 005737 002352  
13 024424 001403  
14 024426 012777 000103 155716  
15 024434 000207  
:RESET  
:RESET ALL UNIBUS DEVICES THEN RESTART THE CLOCK IF IT IS PRESENT  
:INPUTS:  
:   KW.CSR - ADDRESS OF CLOCK REGISTER. ZERO IF NO CLOCK PRESENT.  
:   KWOUT. = DATA TO SEND TO CLOCK REGISTER TO RESTART IT.  
:OUTPUTS:  
:   NONE  
RESET:  BREAK  
        BRESET  
        TST KW.CSR  
        BEQ RESETX  
        MOV #KWOUT.,@KW.CSR  
        ;RESET ALL DEVICES          TRAP  C$BRK  
        ;SEE IF A CLOCK IS PRESENT  TRAP  C$RESET  
        ;START UP THE CLOCK  
RESETX: RETURN
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

024436
024436 010046
024440 010146
024442 010246
024444 010346
024446 010446
024450 010546
024452 005737 002352
024456 001432
024460 013703 002362
024464 013704 002364
024470 013700 002360
024474 004737 020070
024500 012700 000074
024504 004737 020070
024510 010501
024512 004737 020070
024516
024516 010146
024520 010546
024522 010346
024524 012746 006300
024530 012746 000004
024534 010600
024536 104414
024540 062706 000012
024544
024544 012746 005321
024550 012746 000001
024554 010600
024556 104414
024560 062706 000004
024564
024564 012605
024566 012604
024570 012603
024572 012602
024574 012601
024576 012600
024600 000207

```

:RNTIME
:PRINT RUNTIME
:INPUTS:
:   KW.EL - CONTAINS ELAPSED TIME
:   KW.HZ - HERTZ OF CLOCK
:OUTPUTS:
:   THE FOLLOWING IS PRINTED IN BASIC OUTPUT:
:   'RUNTIME HH:MM:SS' AND LINE IS ENDED
RNTIME: PUSH <R0,R1,R2,R3,R4,R5>
      MOV R0,-(SP)
      MOV R1,-(SP)
      MOV R2,-(SP)
      MOV R3,-(SP)
      MOV R4,-(SP)
      MOV R5,-(SP)
      TST KW.CSR           :CHECK IF A CLOCK PRESENT
      BEQ RNTIMX         :BRANCH IF NOT
      MOV KW.EL,R3       :GET ELAPSED TIME
      MOV KW.EL+2,R4
      MOV KW.HZ,R0
      CALL DIVIDE        :GET SPEED OF CLOCK
      MOV #60,R0        :COMPUTE SECONDS OF ELAPSED TIME
      CALL DIVIDE        :NOW DIVIDE BY 60
      MOV R5,R1         :TO COMPUTE MINUTES
      CALL DIVIDE        :SAVE REMAINDER IS SECONDS
      PRINTB #ERRMRT,R3,R5,R1 :DIVIDE BY 60 AGAIN
      :MINUTES IN R5
      MOV R1,-(SP)
      MOV R5,-(SP)
      MOV R3,-(SP)
      MOV #ERRMRT,-(SP)
      MOV #4,-(SP)
      MOV SP,R0
      TRAP C$PNTB
      ADD #12,SP
RNTIMX: PRINTB #ERRNL   :END THE LINE
      MOV #ERRNL,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C$PNTB
      ADD #4,SP
      POP <R5,R4,R3,R2,R1,R0> :HOURS IN R3
      MOV (SP)+,R5
      MOV (SP)+,R4
      MOV (SP)+,R3
      MOV (SP)+,R2
      MOV (SP)+,R1
      MOV (SP)+,R0
      RETURN
      ENDMOD
.SBTTL
    
```

```

1      ;.TITLE MISCELLANEOUS SECTIONS (4)
2      ;SBTTL REPORT CODING SECTION
37
38 024602      BGNMOD
39
40      ;++
41      ; THE REPORT CODING SECTION CONTAINS THE
42      ; 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
43      ;--
44
45 024602      BGNRPT
46
47
48 024602      LSRPT::
49 024602      PUSH <R0,R1,R2,R3,R4,R5>
50 024602      MOV R0,-(SP)
51 024604      MOV R1,-(SP)
52 024606      MOV R2,-(SP)
53 024610      MOV R3,-(SP)
54 024612      MOV R4,-(SP)
55 024614      MOV R5,-(SP)
56
57 024616      PRINTS #RPTMSG,TNUM      ;PRINT TEST NUMBER
58 024616      MOV TNUM,-(SP)
59 024622      MOV #RPTMSG,-(SP)
60 024626      MOV #2,-(SP)
61 024632      MOV SP,R0
62 024634      TRAP C$PNTS
63 024636      ADD #6,SP
64
65 024642      TST KW.CSR      ;CHECK IF A CLOCK IS PRESENT
66 024646      BEQ RPTX      ;BRANCH IF NOT
67 024650      MOV KW.EL,R3      ;GET ELAPSED TIME
68 024654      MOV KW.EL+2,R4
69 024660      MOV KW.HZ,R0
70 024664      CALL DIVIDE
71 024670      MOV #60.,R0      ;GET SPEED OF CLOCK
72 024674      CALL DIVIDE      ;COMPUTE SECONDS OF ELAPSED TIME
73 024700      MOV #60.,R0      ;NOW DIVIDE BY 60
74 024704      CALL DIVIDE      ; TO COMPUTE MINUTES
75 024708      MOV R5,R1      ;SAVE REMAINDER AS SECONDS
76 024712      CALL DIVIDE      ;DIVIDE BY 60 AGAIN
77 024716      PRINTS #RPTMS2,R3,R5,R1      ;MINUTES IN R5, HOURS IN R3
78
79 024706      MOV R1,-(SP)
80 024710      MOV R5,-(SP)
81 024712      MOV R3,-(SP)
82 024714      MOV #RPTMS2,-(SP)
83 024720      MOV #4,-(SP)
84 024724      MOV SP,R0
85 024726      TRAP C$PNTS
86 024730      ADD #12,SP
87
88 024734      RPTX: PRINTS #ERRNL      ;END THE LINE
89 024734      MOV #ERRNL,-(SP)
90 024740      MOV #1,-(SP)
91 024744      MOV SP,R0
92 024746      TRAP C$PNTS
93 024750      ADD #4,SP
94
95 024754      CMP TNUM,#4      ;IF NOT TEST 4
96 024762      BNE RPTXX      ; THAT IS ALL
97 024764      PRINTS #RPTMSH
98 024764      MOV #RPTMSH,-(SP)
99 024770      MOV #1,-(SP)

```



```

025214 016146 000166
025220 016146 000174
025224 012746 002260
025230 012746 025742
025234 012746 000005
025240 010600
025242 104416
025244 062706 000014
108 025250 PRINTS #RPTMD2,D.HERR(R1),D.SERR(R1),D.ECCC(R1)
025250 016146 000176
025254 016146 000172
025260 016146 000170
025264 012746 025772
025270 012746 000004
025274 010600
025276 104416
025300 062706 000012
109 025304 ASSUME D.DRV EQ 0
110 025304 ASSUME C.UADR EQ 0
111 025304 005203 RPTDTN: INC R3 ;COUNT THE DRIVE TABLES
112 025306 020327 000010 CMP R3,#8.
113 025312 002656 BLT RPTDT ;REPEAT FOR ALL DRIVE TABLES
114 025314 062705 000046 RPTCTN: ADD #C.SIZE,R5 ;GO TO NEXT CONTROLLER TABLE
115 025320 005715 TST (R5)
116 025322 001242 BNE RPTCT
117 025324 RPTXX: POP <R5,R4,R3,R2,R1,R0>
025324 012605 MOV (SP)+,R5
025326 012604 MOV (SP)+,R4
025330 012603 MOV (SP)+,R3
025332 012602 MOV (SP)+,R2
025334 012601 MOV (SP)+,R1
025336 012600 MOV (SP)+,R0
118 025340 EXIT RPT
025340 000167 .WORD JSJMP
025342 000450 .WORD L10025-2-.
119
120 025344 DIV10: PUSH R1 ;DIVIDEND IS IN <R2,R3,R4>
025344 010146 MOV R1,-(SP)
121 025346 012701 000060 MOV #48.,R1 ;SET UP SHIFT COUNT
122 025352 005005 CLR R5 ;START WITH ZERO REMAINDER
123 025354 006302 1$: ASL R2 ;SHIFT LEFT INTO R5
124 025356 006103 ROL R3
125 025360 006104 ROL R4
126 025362 006105 ROL R5
127 025364 022705 000012 CMP #10.,R5 ;SILL DIVISOR GO INTO REMAINDER?
128 025370 101003 BHI 2$ ;ONLY SUBTRACT IF IT WILL
129 025372 162705 000012 SUB #10.,R5 ;SUBTRACT DIVISOR
130 025376 005202 INC R2 ;PUT A ONE INTO QUOTIENT
131 025400 005301 2$: DEC R1 ;COUNT THE SHIFTS
132 025402 001364 BNE 1$
133 025404 POP R1 ;RETURN WITH QUOTIENT IN
025404 012601 MOV (SP)+,R1
134 025406 000207 RETURN ; <R2,R3,R4> AND REMAINDER IN R5
135
147 025410 045 116 045 RPTMSG: .ASCIZ\%N%AATEST %D1%A IN PROGRESS.\
148 025444 045 101 040 RPTMS2: .ASCIZ\%A RUNTIME %D3%A:%Z2%A:%Z2\
149 025500 045 116 045 RPTMSH: .ASCIZ\%N%AUNIT DRIVE SERIAL-NUMBER SEEKS MBYTES MBYTES HARD SOFT ECC%N
    
```

150	025615	045	101	040	RPTMH2: .ASCIZ \%
151	025722	045	123	062	RPTMSD: .ASCIZ\%S2%D2%S3%D3%S1\
152	025742	045	124	045	RPTMD1: .ASCIZ\%T%S1%D5%S2%D5%S3%D5%S2\
153	025772	045	104	065	RPTMD2: .ASCIZ\%D5%S2%D5%S1%D5%N\
154					.EVEN
155					
156	026014				ENDRPT
	026014				
	026014	104425			

X1000 READ WRITTEN ERRORS ERRORS%N\

L10025: TRAP CSRPT

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

.SBTTL PROTECTION TABLE

:++
: THIS TABLE IS USED BY THE RUNTIME SERVICES
: TO PROTECT THE LOAD MEDIA.
:--

026016
026016

BGNPROT

L\$PROT::

026016 177777
026020 177777
026022 177777

-1
-1
-1

:OFFSET INTO P-TABLE FOR CSR ADDRESS
:OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
:OFFSET INTO P-TABLE FOR DRIVE NUMBER

026024

ENDPROT

```

1      .SBTTL INITIALIZE SECTION
2
3
4      :++
5      : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
6      : AT THE BEGINNING OF EACH PASS.
7      :--
8 026024      BGNINIT
9
10      000000
11
12      .REPT 0
13      IF HERE FROM CONTINUE COMMAND
14      THEN
15          SET ICONT BIT IN IFLAGS
16      ENDIF
17      IF HERE FROM RESTART COMMAND
18      THEN
19          SET IREST BIT IN IFLAGS
20      ENDIF
21      IF HERE FROM POWER FAIL RESTART
22      THEN
23          RESET ALL UNITS
24          INITIALIZE MEMORY MANAGEMENT REGISTERS
25          PRINT STATISTICAL REPORT
26          DELAY ONE MINUTE
27      ENDIF
28      IF HERE FROM START COMMAND
29      THEN
30          RESET ALL UNITS
31          INITIALIZE MEMORY MANAGEMENT REGISTERS
32          ESTABLISH FREE MEMORY
33          CLEAR TNUM
34          SET ISTRT BIT IN IFLAGS, CLEAR OTHER BITS
35          INITIALIZE CLOCK
36          BUILD TABLES
37      ELSE
38          CHECK TABLES FOR ADDED OR DROPPED UNITS
39      .ENDIF
40      .ENDR
41
42      READEF #EF.CONTINUE      ;HERE FROM CONTINUE COMMAND?
43      MOV #EF.CONTINUE,RO
44      TRAP C$REFG
45
46      BNCOMPLETE INIT1      ;JUMP IF NOT
47      BCC INIT1
48      ;SET CONTINUE BIT IN FLAG RECORD
49      ;LOOK AT EVENT FLAGS
50      MOV #EF.RESTART,RO
51      TRAP C$REFG
52
53      BNCOMPLETE INIT1A      ;SET IREST BIT IN IFLAGS
54      BCC INIT1A
55      ; IF HERE FROM RESTART COMMAND
56      ;HERE FROM POWER RESTART?
57      MOV #EF.PWR,RO
58      TRAP C$REFG
59
60      BNCOMPLETE INIT2      ;JUMP IF SET
61      BCC INIT2
62
63 026024      012700 000036
64 026030      104447
65 026032      103003
66 026034      052737 000002 002224
67 026042      012700 000037
68 026046      104447
69 026050      103003
70 026052      052737 000004 002224
71 026060      012700 000034
72 026064      104447
73 026066      103052
  
```

```

71 026070 004737 024414          CALL RESET          ;RESET ALL UNITS
72 026074 004737 022240          CALL INITKT         ;INITIALIZE MEMORY MANAGEMENT REGISTERS
73 026100          104424          DORPT              ;PRINT A STATISTICAL REPORT
                                TRAP      C$DRPT
74 026102 005737 002352          TST KW.CSR         ;CHECK IF CLOCK ON SYSTEM
75 026106 001421          BEQ POWDLY         ;IF NOT, USE A LOOP DELAY
76 026110 012700 000074          MOV #60.,R0        ;SET UP A TIME OUT OF 60 SECONDS
77 026114 012701 002372          MOV #STIME,R1
78 026120 004737 022660          CALL SETTO
79 026124          104422          POWDLC: BREAK
                                TRAP      C$BRK
80 026126 023737 002374 002364          CMP STIME+2,KW.EL+2 ;CHECK IF TIME ELAPSED
81 026134 101027          BHI INIT2
82 026136 001372          BNE POWDLC
83 026140 023737 002372 002362          CMP STIME,KW.EL
84 026146 103766          BLO POWDLC
85 026150 000421          BR INIT2
86 026152 012701 002400          POWDLY: MOV #2400,R1 ;DELAY ONE MINUTE FOR DISK
87 026156          012727 000372          POWDLL: DELAY 250. ; UNITS TO POWER UP
                                MOV      #250.,(PC)+
                                .WORD   0
                                MOV      L$DLY,(PC)+
                                .WORD   0
                                DEC      -6(PC)
                                BNE     -.4
                                DEC     -22(PC)
                                BNE     .-20
88 026206          104422          BREAK              ; WHILE RESPONDING TO ^C
                                TRAP      C$BRK
89 026210 005301          DEC R1              ; EVERY0.025 SECOND
90 026212 003357          BGT POWDLY
91 026214          012700 000040          INIT2: READEF #EF.START ;HERE FROM START COMMAND?
                                MOV      #EF.START,R0
                                TRAP      C$REFG
92 026222          103467          BCOMPLETE INIT3    ;JUMP IF NOT
                                BCS      INIT3
93
94          ;SET NOT AVAILABLE BITS IN ALL CONTROLLER AND DRIVE TABLES
95
96 026224 013705 002204          MOV CTABS,R5        ;GET FIRST CONTROLLER TABLE ADDRESS
97
98 026230 052765 100000 000002          INITC1: BIS #CT.AVL,C.UNIT(R5) ;SET BIT IN CONTROLLER TABLE
99 026236 010502          MOV R5,R2          ;GET POINTER TO DRIVE TABLES
100 026240 062702 000020          ADD #C.DR0,R2
101 026244 012703 000010          MOV #8.,R3         ;GET COUNT OF DRIVE TABLES
102 026250 012200          INITC2: MOV (R2)+,R0 ;CHECK IF ANY MORE DRIVE TABLES
103 026252 001405          BEQ INITC3
104 026254 052760 100000 000002          BIS #DT.AVL,D.UNIT(R0) ;SET BIT IN DRIVE TABLE
105 026262 005303          DEC R3
106 026264 003371          BGT INITC2
107 026266 062705 000046          INITC3: ADD #C.SIZE,R5 ;MOVE TO NEXT CONTROLLER TABLE
108 026272 005715          TST (R5)           ;IS THERE A NEXT ONE?
109 026274 001355          BNE INITC1        ;IF SO, CLEAR THE BITS THERE
110
111          ;NOW GET EACH P-TABLE AND CLEAR NOT AVAILABLE BITS
112
113 026276 005003          CLR R3             ;START WITH UNIT 0
  
```



```

114 026300          INITC4: GPHARD R3,R0          ;GET HW P-TABLE
      026300 010300          MOV R3,R0
      026302 104442          TRAP C$GPHRD
115 026304          BNCOMPLETE INITC7           ;GO AROUND IF NOT AVAILABLE
      026304 103030          BCC INITC7
116 026306 013705 002204   INITC5: MOV CTABS,R5          ;GET FIRST CONTROLLER TABLE
117 026312 021015          CMP (R0),(R5)         ;COMPARE UDA ADDRESSES
118 026314 001411          BEQ INITCC
119 026316 062705 000046   ADD #C.SIZE,R5          ;LOOK AT NEXT CONTROLLER TABLE
120 026322 005715          TST (R5)             ;IF THERE IS ANY
121 026324 001372          BNE INITC5
122 026326          ERRSF 1,BADT
      026326 104454          TRAP C$ERSF
      026330 000001          .WORD 1
      026332 003513          .WORD BADT
      026334 000000          .WORD 0
123 026336          INITE1: DOCLN              TRAP C$DCLN
      026336 104444
124 026340 016001 000010   INITCC: MOV HO.LDR(R0),R1
125 026344 004737 017414   CALL GTDRVT
126 026350 001372          BNE INITE1
127 026352 042765 100000 000002   INITC6: BIC #CT.AVL,C.UNIT(R5)
128 026360 042764 100000 000002   BIC #DT.AVL,D.UNIT(R4)
129 026366 005203          INITC7: INC R3
130 026370 023703 002012   CMP L$UNIT,R3
131 026374 003341          BGT INITC4
132 026376 000137 027412   JMP INITXX

```

```

1 026402          INIT3: BRESET          ;RESET ALL UNITS
   026402 104433          TRAP          C$RESET
2 026404 004737 022240          CALL INITKT          ;INITIALIZE MEMORY MANAGEMENT
3 026410          MEMORY FFREE          ;RESET START OF FREE MEMORY
   026410 104431          TRAP          C$MEM
   026412 010037 002174          MOV          RO,FFREE
4 026416 017737 153552 002176          MOV @FFREE,FSIZE          ;RESET SIZE OF FREE MEMORY
5 026424 005037 002226          CLR TNUM          ;INITIALIZE TEST NUMBER TO NO TEST RUNNING
6 026430 012737 000010 002224          MOV #ISTRT,IFLAGS          ;SET START FLAG FOR TEST 4
7
8          ;INITIALIZE CLOCK
9
10          KWOUT.=103          ;DATA TO SEND TO KW11 TO START CLOCK
11 026436 005037 002362          CLR KW.EL          ;CLEAR ELAPSED TIME
12 026442 005037 002364          CLR KW.EL+2
13 026446          CLOCK L,RO          ;SEE IF AN L CLOCK PRESENT
   026446 012700 000114          MOV          #'L,RO
   026452 104462          TRAP          C$CLK
14 026454          BCOMPLETE KYES
   026454 103417          BCS          KYES
15 026456          CLOCK P,RO          ;SEE IF A P CLOCK PRESENT
   026456 012700 000120          MOV          #'P,RO
   026462 104462          TRAP          C$CLK
16 026464          BCOMPLETE KYES
   026464 103413          BCS          KYES
17 026466 005037 002352          CLR KW.CSR          ;IF NEITHER, CLEAR CSR STORAGE WORD
18 026472          PRINTF #NOCLOCK
   026472 012746 011044          MOV          #NOCLOCK,-(SP)
   026476 012746 000001          MOV          #1,-(SP)
   026502 010600          MOV          SP,RO
   026504 104417          TRAP          C$PNTF
   026506 062706 000004          ADD          #4,SP
19 026512 000444          BR KNO
20 026514 012037 002352          KYES: MOV (RO)+,KW.CSR          ;STORE DATA RETURNED
21 026520 012037 002354          MOV (RO)+,KW.BRL
22 026524 012037 002356          MOV (RO)+,KW.VEC
23 026530 012037 002360          MOV (RO)+,KW.HZ
24 026534          SETVEC KW.VEC,#KW11I,KW.BRL          ;SET THE VECTOR
   026534 013746 002354          MOV          KW.BRL,-(SP)
   026540 012746 024364          MOV          #KW11I,-(SP)
   026544 013746 002356          MOV          KW.VEC,-(SP)
   026550 012746 000003          MOV          #3,-(SP)
   026554 104437          TRAP          C$SVEC
   026556 062706 000010          ADD          #10,SP
25 026562 012777 000103 153562          MOV #KWOUT,@KW.CSR          ;START THE CLOCK
26 026570 013702 002360          MOV KW.HZ,R2          ;INITIALIZE TIME FOR FIRST
27 026574 000302          SWAB R2          ; STATISTICAL REPORT
28 026576 006302          ASL R2          ; FOR ABOUT 15 MINUTES FROM NOW
29 026600 006302          ASL R2
30 026602 063702 002362          ADD KW.EL,R2
31 026606 013737 002364 002374          MOV KW.EL+2,STIME+2
32 026614 010237 002372          MOV R2,STIME
33 026620 005537 002374          ADC STIME+2
34 026624          KNO:

```

```

1           ;INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
2
3 026624 013737 002174 002204      MOV FFREE,CTABS      ;STORE START OF CONTROLLER TABLES
4 026632 005077 153346              CLR @CTABS          ;ZEROS MARKS END CONTROLLER TABLES
5 026636 005037 002206              CLR CTRLRS         ;CLEAR CONTROLLER COUNT
6
7           ;GET A P-TABLE FROM DRS
8
9 026642 005002                      CLR R2              ;LOGICAL UNIT NUMBER IN R2
10 026644 010200                      INIT4: GPHARD R2,R0 ;GET POINTER TO A P-TABLE
    026644 010200                                MOV R2,R0
    026646 104442                                TRAP C$GPHRD
11 026650                                BNCOMPLETE NXTTAB ;IGNORE IF NO TABLE RETURNED
    026650 103064                                BCC NXTTAB
12
13          ;SEE IF A CONTROLLER TABLE ALREADY EXISTS FOR CONTROLLER IN P-TABLE
14
15 026652 013703 002204              MOV CTABS,R3       ;GET ADDRESS OF CONTROLLER TABLES
16 026656 005713                      INIT5: TST (R3)     ;CHECK IF ANY MORE TABLES
17 026660 001405                      BEQ NEWTAB         ;BUILD NEW TABLE IF FOUND ZERO WORD
18 026662 021013                      CMP (R0),(R3)      ;CHECK IF SAME UNIBUS ADDRESS
19 026664                                ASSUME C.UADR EQ 0
20 026664                                ASSUME HO.UBA EQ 0
21 026664 001437                      BEQ SAMTAB         ;CHECK TABLE IF ALREADY EXISTS
22 026666 062703 000046              ADD #C.SIZE,R3    ;MOVE TO NEXT TABLE
23 026672 000771                      BR INIT5

```

```
1  
2 ;BUILD A CONTROLLER TABLE  
3 026674 012701 000023 NEWTAB: MOV #C.SIZE/2,R1 ;GET WORDS IN CONTROLLER TABLE  
4 026700 004737 012722 CALL ALOCM ;ALLOCATE SPACE FOR IT  
5 026704 011021 MOV (R0),(R1)+ ;STORE UNIBUS ADDRESS  
6 026706 010221 MOV R2,(R1)+ ;UNIT NUMBER  
7 026710 016004 000004 MOV HO.BRL(R0),R4 ;GET BR LEVEL  
8 026714 000304 SWAB R4 ;SWAP TO HIGH BYTE  
9 026716 006104 ROL R4 ;SHIFT ONE MORE TO LEFT  
10 026720 056004 000002 BIS HO.VEC(R0),R4 ;ADD VECTOR ADDRESS  
11 026724 010421 MOV R4,(R1)+ ; TO TABLE  
12 026726 016021 000006 MOV HO.BST(R0),(R1)+  
13 026732 012721 004037 MOV #4037,(R1)+ ;PUT [JSR R0,UDASRV]  
14 026736 012721 022650 MOV #UDASRV,(R1)+ ; INTO TABLE  
15 026742 012703 000015 MOV #13.,R3 ;CLEAR POINTERS TO DRIVE TABLES,  
16 026746 005021 INIT7: CLR (R1)+ ; TIMEOUT COUNTER, FLAGS, REF. NUMBER  
17 026750 005303 DEC R3  
18 026752 001375 BNE INIT7 ;LOOP TIL ALL CLEARED  
19 026754 005237 002206 INC CTRLRS ;COUNT THE CONTROLLER  
20 026760 005011 CLR (R1) ;CLEAR TABLE END MARKER  
21 026762 000417 BR NXTTAB ;NOW GO TO NEXT P-TABLE
```

```

1          ;SHOULD BE SAME CONTROLLER, CHECK THAT OTHER PARAMETERS MATCH
2
3 026764 016004 000004      SAMTAB: MOV HO.BRL(R0),R4          :GET BR LEVEL FROM P-TABLE
4 026770 000304              SWAB R4                    :SWAP TO HIGH BYTE
5 026772 006104              ROL R4                      :SHIFT ONE MORE TO LEFT
6 026774 056004 000002      BIS HO.VEC(R0),R4          :ADD VECTOR ADDRESS
7 027000 020463 000004      CMP R4,C.VEC(R3)          :COMPARE WITH CONTROLLER TABLE
8 027004 001004              BNE 1$
9 027006 026063 000006 000006  CMP HO.BST(R0),C.BST(R3)      :COMPARE BURST RATES
10 027014 001402              BEQ NXTTAB
11 027016 000137 027434      1$: JMP CTABER              :FATAL ERROR IF NOT SAME
12
13          ;GET NEXT P-TABLE
14
15 027022 005202              NXTTAB: INC R2              :INCREMENT LOGICAL UNIT NUMBER
16 027024 023702 002012      CMP L$UNIT,R2          :CHECK IF GOT ALL TABLES
17 027030 003305              BGT INIT4                :IF NOT, GO BACK FOR NEXT
18
19 027032 012701 000001      MOV #1,R1              :ALLOCATE SPACE FOR ZERO END WORD
20 027036 004737 012722      CALL ALOCM          :AFTER CONTROLLER TABLES

```

```
1  
2 ;NOW BUILD DRIVE TABLES  
3 027042 005005 CLR R5 ;CLEAR CUSTOMER DATA FLAG  
4 027044 005002 CLR R2 ;LOGICAL UNIT NUMBER IN R2  
5 027046 INIT8: GPHARD R2,R0 ;GET POINTER TO A P-TABLE  
6 027050 010200 ; ;MOV R2,R0  
7 027052 104442 ; ;TRAP C$GPHRD  
8 027052 103056 BNCOMPLETE INIT14 ;IF NOT AVAILABLE, GO GET NEXT  
9 ; ;BCC INIT14  
10 ;FIND CONTROLLER TABLE  
11 027054 013703 002204 INIT10: MOV CTABS,R3 ;GET ADDRESS OF CONTROLLER TABLES  
12 027060 021013 CMP (R0),(R3) ;CHECK IF SAME UNIBUS ADDRESS  
13 027062 001403 BEQ INIT11 ;BRANCH IF TABLE FOUND  
14 027064 062703 000046 ADD #C.SIZE,R3 ;MOVE TO NEXT TABLE  
15 027070 000773 BR INIT10
```

```

1      ;BUILD DRIVE TABLE
2
3 027072 012701 000103   INIT11: MOV #D.SIZE/2,R1           ;GET SIZE OF DRIVE TABLE
4 027076 004737 012722   CALL ALOCM                       ;ALLOCATE SPACE FROM FREE MEMORY
5
6      : R0 POINTS TO P-TABLE
7      : R1 POINTS TO DRIVE TABLE
8      : R3 POINTS TO CONTROLLER TABLE
9      : R2 IS LOGICAL UNIT NUMBER
9 027102 062703 000020   ADD #C.DR0,R3                     ;BUILD POINTER TO C.DR ENTRY IN CONTROLLER TABLE
10 027106 012704 000010   MOV #8.,R4                         ;GET MAX COUNT OF DRIVES ON ONE CONTROLLER
11 027112 005713         INIT12: TST (R3)                    ;CHECK IF ENTRY CONTAINS POINTER TO DRIVE TABLE
12 027114 001411         BEQ INIT13
13 027116 026033 000010   CMP HO.LDR(R0),@(R3)+              ;CHECK DRIVE NUMBER IN DRIVE TABLE
14 027122 001002         BNE 1$
15 027124 000137 027446   JMP MLDRER                           ;IF SAME, TWO P-TABLES POINT TO SAME DRIVE
16 027130 005304         1$: DEC R4                          ;COUNT DRIVES
17 027132 001367         BNE INIT12                       ;IF EIGHT DRIVE TABLES EXIST,
18 027134 000137 027460   JMP TOOMER                           ; THEN REPORT ERROR
19 027140 010113         INIT13: MOV R1,(R3)                ;LOAD DRIVE TABLE POINTER
20 027142 016021 000010   MOV HO.LDR(R0),(R1)+               ;LOAD DRIVE NUMBER
21 027146 010221         MOV R2,(R1)+                          ;LOAD UNIT NUMBER
22 027150 016011 000012   MOV HO.PRM(R0),(R1)                ;GET TEST AREA BIT
23 027154 051105         BIS (R1),R5                          ;SAVE 'DR' OF BIT FROM ALL DRIVES
24 027156 005111         COM (R1)                            ;COMPLIMENT IT
25 027160         AND HM.CYL,(R1)
26 027164 042711 157777   BIC #^C<HM.CYL>,(R1)
27 027170 012703 000100   BIS #DDEF,(R1)+                    ;LOAD DEFAULT PARAMETER BITS
28 027174 005021         MOV #<D.SIZE/2-3>,R3                ;CLEAR REST OF TABLE
29 027176 005303         INIT3L: CLR (R1)+
30 027200 003375         DEC R3
31 027202 012761 177777 177754 BGT INIT3L
31 027202 012761 177777 177754 MOV #-1,<D.ECYL+2-D.SIZE>(R1) ;MARK CYLINDERS AT TEST ALL
    
```

```

1          ;GO TO NEXT DRIVE TABLE
2
3 027210 005202
4 027212 023702 002012
5 027216 003313
6
7          ;IF ANY DRIVE SELECTED FOR EXERCISE IN CUSTOMER DATA AREA
8          ;GIVE WARNING
9
10 027220 032705 020000
11 027224 001464
12 027226
    027226 012746 007053
    027232 012746 000001
    027236 010600
    027240 104417
    027242 062706 000004
13 027246 013705 002204
14 027252 010504
15 027254 062704 000020
16 027260 012701 000010
17 027264 012403
18 027266 001422
19 027270 032763 020000 000004
20 027276 001014
21 027300
    027300 011346
    027302 011546
    027304 016346 000002
    027310 012746 007165
    027314 012746 000004
    027320 010600
    027322 104417
    027324 062706 000012
22 027330
23 027332 001354
24 027334 062705 000046
25 027340 005715
26 027342 001343
27
28
29
30 027344
    027344 104450
31 027346
    027346 103013
32 027350
    027350 104443
    027352 000404
    027354 002260
    027356 000120
    027360 005171
    027362 000001
    027364
33 027364 032737 000001 002260
34 027372 001001
35 027374

```

```

;INCREMENT LOGICAL UNIT NUMBER
;CHECK IF GOT ALL TABLES
;IF NOT, GET NEXT TABLE

;CHECK IF BIT EVER SET
;BYPASS IF NOT
;PRINT WARNING HEADER
MOV #INITWA,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD #4,SP

;GET FIRST CONTROLLER TABLE
;GET ADDRESS OF POINTER TO DRIVE TABLE
MOV CTABS,R5
MOV R5,R4
ADD #C.DR0,R4
MOV #8,R1

;GET COUNT OF DRIVE TABLES
;GET ADDRESS OF DRIVE TABLE
MOV (R4)+,R3
BEQ INITW4
BIT #D.DCY,D.PRM(R3)
BNE INITW3
PRINTF #INITWB,D.UNIT(R3),(R5),(R3) ;PRINT NUMBERS
MOV (R3)-,(SP)
MOV (R5)-,(SP)
MOV D.UNIT(R3)-,(SP)
MOV #INITWB,-(SP)
MOV #4,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD #12,SP

;COUNT THE DRIVE TABLES
;LOOK AT ALL OF THEM
;MOVE TO NEXT CONTROLLER TABLE
;SEE IF ANOTHER TABLE AND
; LOOK AT IT
DEC R1
BNE INITW2
ADD #C.SIZE,R5
TST (R5)
BNE INITW1

;CHECK IF MANUAL INTERVENTION ALLOWED
TRAP C$MANI

;BRANCH IF ALLOWED
BCC INIT15

;ASK OPERATOR
TRAP C$GMAN
BR 10000$
.WORD TEMP
.WORD T$CODE
.WORD INITWC
.WORD 1
10000$:

;LOOK AT RESPONSE
;BRANCH IF YES WAS ANSWER
;ABORT PROGRAM
DOCLN

```



```

027374 104444                                TRAP  C$DCLN
36
37 ;SAVE CURRENT PARAMETERS TO FREE MEMORY SO EACH TEST CAN USE ALL OF IT
38
39 027376 013737 002174 002200 INIT15: MOV FFREE,FMEM          ;SAVE START ADDRESS
40 027404 013737 002176 002202          MOV FSIZE,FMEMS        ;SAVE SIZE
41
42 027412                                INITXX: SETPRI #PRI00    ; SET RUNNING PRIORITY TO ZERO
   027412 012700 000000                                MOV #PRI00,R0
   027416 104441                                TRAP C$SPRI
43 027420 005037 002412                                CLR DLL                ;ERASE DOWNLINE LOAD DATA
44 027424 004737 023716                                CALL CLOSEF            ;MAKE SURE DATA FILE IS CLOSED
45 027430                                KPRI: EXIT INIT
   027430 104432                                TRAP  C$EXIT
   027432 000040                                .WORD L10027-.

```

```

1          ;DIFFERENT VECTORS, BR LEVELS OR BURST RATES FOR ONE CONTROLLER
2 027434   CTABER: ERRSF 1,,CTABE
   027434 104454
   027436 000001
   027440 000000
   027442 012612
3 027444   DOCLN
   027444 104444
4
5          ;TWO P-TABLES FOR SAME DRIVE
6 027446   MLDRE: ERRSF 1,MLDREM
   027446 104454
   027450 000001
   027452 003370
   027454 000000
7 027456   DOCLN
   027456 104444
8
9          ;MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER
10
11 027460   TOOMER: ERRSF 1,TOOMEM
   027460 104454
   027462 000001
   027464 003431
   027466 000000
12 027470   DOCLN
   027470 104444
13
14
15 027472   ENDINIT
   027472
   027472 104411

```

L10027: TRAP C\$INIT

1
2
3
4
5
6
7
8
9

.SBTTL AUTODROP SECTION

;++
: THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
: THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
: SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
: DROPPED FROM TESTING.
:--

10 027474
027474

BGNAUTO

LSAUTO::

11
18 027474
027474
027474 104461

ENDAUTO

L10030: TRAP CSAUTO

```
1          .SBTTL  CLEANUP CODING SECTION
2
3
4          :++
5          : THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
6          : AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
7          :--
8 027476          BGNCLN
9 027476
18 027476 004737 023716          CALL CLOSEF
30 027502          ENDCLN          ;CLOSE DATA FILE
   027502
   027502 104412          L10031: TRAP  C$CLEAN
31 027504          ENDMOD
32          .SBTTL
```

```

1          ;.TITLE HARDWARE TESTS (5)
9
10         .SBTTL TEST 1: UNIBUS ADDRESSING TEST
15
16 027504      BGNMOD
17
18 027504      BGNST
19 027504      012701 000001      MOV #1,R1      ; INITIALIZE TEST PARAMETERS
20 027510      004737 013006      CALL TINIT
21 027514      013737 002204 002210      MOV CTABS,TSTTAB ; GET ADDRESS OF FIRST CONTROLLER TABLE
22 027522      013705 002210      T1NEXT: MOV TSTTAB,R5 ; GET CONTROLLER TABLE ADDRESS
23 027526      016537 000002 002074      MOV C.UNIT(R5),L$LUN ; CHECK IF UNIT AVAILABLE FOR TESTING
24 027534      100504      BMI T1SKIP ; SKIP TESTING IF NOT AVAILABLE
25 027536      ASSUME CT.AVL EQ BIT15
26
27 027536      004737 024414      CALL RESET ; RESET ALL UNITS
28 027542      BGNSUB; 1
29 027542      104402      T1.1: TRAP C$BSUB
30 027544      005037 002366      CLR NXMAD ; CLEAR MEMORY ERROR FLAG
31 027550      012746 000340      SETVEC #4,#NXMI,#PRI07
32 027554      012746 022640      MOV #PRI07,-(SP)
33 027560      012746 000004      MOV #NXMI,-(SP)
34 027564      012746 000003      MOV #4,-(SP)
35 027570      104437      MOV #3,-(SP)
36 027572      062706 000010      TRAP C$SVEC
37 027576      011504      ADD #10,SP
38 027600      005714      MOV (R5),R4 ; GET ADDRESS OF UDAIP REGISTER
39 027602      005764 000002      TST (R4) ; READ UDAIP
40 027606      012700 000004      TST 2(R4) ; READ UDASA
41 027612      104436      CLRVEC #4 ; GIVE UP VECTOR
42 027614      005737 002366      MOV #4,R0
43 027620      001405      TST NXMAD ; CHECK FLAG
44 027622      104455      TRAP C$CVEC
45 027624      000025      TRAP C$ERDF
46 027626      000000      .WORD 21
47 027630      012154      .WORD 0
48 027632      000445      .WORD INTR4
49 027634      BR T1SKIP ; DON'T TEST ANY MORE
50 027634      T1GOOD:
51 027634      ENDSUB
52 027634      104403      L10033: TRAP C$ESUB
53 027636      BGNSUB; 2
54 027636      104402      T1.2: TRAP C$BSUB
55 027640      004737 027764      CALL PORTST ; DIAGNOSTIC LOOP MODE TESTING
56 027644      ENDSUB
57 027644      104403      L10034: TRAP C$ESUB
58 027646      BGNSUB; 3
59 027646      104402      T1.3: TRAP C$BSUB
60 027650      004737 030234      CALL INTEST ; TEST INTERRUPTS

```

```

46 027654          ENDSUB
   027654
   027654 104403          L10035:          TRAP    C$ESUB
47 027656          BGNSUB; 4
   027656          T1.4:          TRAP    C$BSUB
   027656 104402          ; INITIALIZE UDA WITH SMALLEST
48 027660 005004          CLR    R4          ; RING BUFFER AND INTERRUPTS DISABLED
49 027662 004737 022742  CALL   UDAINT
50 027666          ENDSUB
   027666          L10036:          TRAP    C$ESUB
   027666 104403          T1.5:          TRAP    C$BSUB
51 027670          BGNSUB; 5
   027670          ; INITIALIZE UDA WITH RING BUFFER LARGE
   027670 104402          MOV    #132000,R4  ; ENOUGH TO COVER NORMAL HOST COMM AREA
52 027672 012704 132000  CALL   UDAINT      ; PACKET AND BUFFER SPACE (A 5 IN MES
53 027676 004737 022742          ; LENGTH AND A 5 IN CMD LENGTH)
54
55
56 027702          ENDSUB
   027702          L10037:          TRAP    C$ESUB
   027702 104403          T1.6:          TRAP    C$BSUB
57 027704          BGNSUB; 6
   027704          ; SAVE FREE MEMORY PARAMETERS
   027704 104402          PUSH   FFREE
58 027706 013746 002174  MOV   FFREE,-(SP)
   027706          PUSH   FSIZE
59 027712 013746 002176  MOV   FSIZE,-(SP)
   027712          MOV    #1,R1          ; RUN DM PROGRAM IN
60 027716 012701 000001  CALL   RUNDM      ; ONE CONTROLLER ONLY
61 027722 004737 013134  BEQ  1$
62 027726 001402          CALL  RESPDM
63 027730 004737 013224  POP    FSIZE
64 027734          1$:          MOV (SP)+,FSIZE
   027734 012637 002176  POP    FFREE
65 027740          POP    FFREE
   027740 012637 002174  MOV (SP)+,FFREE
66 027744          ENDSUB
   027744          L10040:          TRAP    C$ESUB
   027744 104403

```

TEST 1: UNIBUS ADDRESSING TEST

1	027746	062737	000046	002210	T1SKIP:	ADD	#C.SIZE, TSTTAB	:	MOVE TO NEXT CONTROLLER
2	027754	005777	152230			TST	@TSTTAB	:	CHECK IF ANOTHER CONTROLLER TABLE
3	027760	001260				BNE	T1NEXT		
4									
5	027762				T1ERR:				
6	027762					ENDTST			
	027762							L10032:	
	027762	104401						TRAP	CSETST

```

1 027764          PORTST:
2
3
4
5 027764 011504          MOV      (R5),R4          ; R4 POINTS TO UDAIP REGISTER
6 027766          ASSUME C.UADR EQ 0
7 027766 005014          CLR      (R4)           ; INITIALIZE THE UDA
8 027770 012737 004000 023652  MOV      #SA.S1,UDARSD   ; LOOK FOR STEP 1
9 027776 004737 023514          CALL     UDARSP          ; WAIT FOR RESPONSE
10 030002 103453          BCS     3$              ; IF ERROR, BRANCH
11 030004 016437 000002 030232  MOV      2(R4),WCHNGD    ; MOVE OLD PORT CONTENTS TO STORAGE
12 030012 012764 140000 000002  MOV      #140000,2(R4)   ; INITIALIZE FOR PORT WRAP
13 030020 004737 030136          CALL     WCHNG          ; WAIT FOR THE PORT TO CHANGE
14 030024 001442          BEQ     3$              ; IF ERROR, BRANCH
15 030026 022764 140000 000002  CMP      #140000,2(R4)   ; COMPARE WITH DATA WRITTEN
16 030034 001405          BEQ     4$              ; IF NO ERROR, BRANCH
17 030036          ERRDF   58,LOOP02,LOOPA ; REPORT ERROR
    030036 104455          TRAP    C$ERDF
    030040 000072          .WORD  58
    030042 005012          .WORD  LOOP02
    030044 012534          .WORD  LOOPA
18 030046 000431          BR      3$              ; EXIT, REPORTING ERROR TO CALLING ROUTINE
19 030050 012702 000001          4$:    MOV      #1,R2          ; SET UP FOR SHIFTING '1'
20 030054 012703 000020          MOV      #16.,R3        ; SET UP LOOP COUNT
21 030060 016437 000002 030232  1$:    MOV      2(R4),WCHNGD    ; SAVE OLD PORT CONTENTS
22 030066 010264 000002          MOV      R2,2(R4)       ; WRITE PATTERN TO UDASA FOR LOOP
23 030072 004737 030136          CALL     WCHNG          ; WAIT FOR UDASA TO CHANGE
24 030076 001415          BEQ     3$              ; IF ERROR, BRANCH
25 030100 020264 000002          CMP      R2,2(R4)       ; COMPARE R0 WITH WHAT WAS ECHOED
26 030104 001405          BEQ     2$              ; IF MATCH, BRANCH
27 030106          ERRDF   57,LOOP02,LOOP ; REPORT ERROR
    030106 104455          TRAP    C$ERDF
    030110 000071          .WORD  57
    030112 005012          .WORD  LOOP02
    030114 012504          .WORD  LOOP
28 030116 000405          BR      3$              ; BRANCH
29 030120 006302          2$:    ASL     R2              ; MOVE THE SHIFTING ONE LEFT BY 1
30 030122 005303          DEC     R3              ; DECREMENT COUNT
31 030124 001355          BNE     1$              ; IF LOOP INCOMPLETE, BRANCH
32 030126 000244          CLZ                    ; FLAG AS NO ERRORS
33 030130 000207          RETURN                 ; RETURN TO TEST 1 CODE
34 030132 000264          3$:    SEZ                    ; FLAG AS ERROR OCCURRED
35 030134 000207          RETURN                 ; RETURN TO TEST 1 CODE
    
```



```

1          ;WCHNG
2          ;
3          ;
4          ;      WAIT UNTIL UDASA CHANGES FROM WHAT IS IN WCHNGD
5 030136 012700 000012      WCHNG:  MOV #10.,R0          ;SET TIMEOUT FOR 10 SECONDS
6 030142 010501          MOV R5,R1          ;POINT TO CONTROLLER TABLE
7 030144 062701 000040      ADD #C.TO,R1
8 030150 004737 022660      CALL SETTO
9 030154 026437 000002 030232 1$:  CMP 2(R4),WCHNGD      ;SEE IF CHANGED
10 030162 001022          BNE 2$
11 030164          BREAK
12 030164 104422          TST KW.CSR          TRAP      C$BRK
13 030166 005737 002352          ;SEE IF CLOCK ON SYSTEM
14 030172 001770          BEQ 1$
15 030174 023765 002364 000042      CMP KW.EL+2,C.TO(R5)      ;CHECK IF TIME OUT OCCURRED
16 030202 101005          BHI 3$
17 030204 001363          BNE 1$
18 030206 023765 002362 000040      CMP KW.EL,C.TO(R5)
19 030214 103757          BLO 1$
20 030216          3$:  ERRDF 56,LOOP01      ; REPORT ERROR
21 030216 104455          TRAP      C$ERDF
22 030220 000070          .WORD 56
23 030222 004715          .WORD LOOP01
24 030224 000000          .WORD 0
25 030226 000264          SEZ          ; FLAG AS ERROR
26 030230 000207          2$:  RETURN      ; RETURN TO CALLING PROGRAM
27
28
29
30 030232          WCHNGD: .BLKW 1      ; OLD PORT CONTENTS

```

```

1 030234          INTEST:
2                :
3                :
4                :   TEST THE INTERRUPTS VECTOR AND BR LEVEL
5 030234 011504   MOV      (R5),R4           ; R4 POINTS TO UDAIP REGISTER
6 030236          ASSUME C.UADR EQ 0
7 030236 016503 000004   MOV      C.VEC(R5),R3       ; GET VECTOR AND BRANCH LEVEL
8 030242 010302   MOV      R3,R2           ; COPY TO R2 FOR BR LEVEL
9 030244 042703 177000   BIC     #^CCT.VEC,R3       ; CLEAR UNUSED VECTOR BITS
10 030250 042702 170777  BIC     #^CCT.BRL,R2      ; CLEAR UNUSED BRANCH LEVEL BITS
11 030254 012701 000011   MOV     #9.,R1           ; SET UP TO SHIFT BR LEVEL
12 030260 006202          1$:  ASR     R2           ; SHIFT BY ONE BIT
13 030262 005301          DEC     R1           ; COUNT SHIFTS
14 030264 001375          BNE    1$           ; IF INCOMPLETE, BRANCH
15 030266 010237 030706   MOV     R2,BRLEV        ; SAVE THE BRANCH LEVEL
16 030272          PRINTF #INTST0,C.UADR(R5),R3 ; PRINT BEGINNING OF INTERRUPT MESSAGE
    030272 010346          MOV     R3,-(SP)
    030274 016546 000000          MOV     C.UADR(R5),-(SP)
    030300 012746 006672          MOV     #INTST0,-(SP)
    030304 012746 000003          MOV     #3,-(SP)
    030310 010600          MOV     SP,R0
    030312 104417          TRAP   C$PNTF
    030314 062706 000010          ADD     #10,SP
17 030320          SETVEC R3,#INTSRV,#PRI00 ; SET UP INTERRUPT ROUTINE
    030320 012746 000000          MOV     #PRI00,-(SP)
    030324 012746 024406          MOV     #INTSRV,-(SP)
    030330 010346          MOV     R3,-(SP)
    030332 012746 000003          MOV     #3,-(SP)
    030336 104437          TRAP   C$SVEC
    030340 062706 000010          ADD     #10,SP
18 030344          SETPRI #PRI00           ; SET PRIORITY TO 0 TO CHECK INTERRUPTS
    030344 012700 000000          MOV     #PRI00,R0
    030350 104441          TRAP   C$SPRI
19 030352 006203          ASR     R3           ; DIVIDE VECTOR BY 4 FOR UDA INITIALIZATION
20 030354 006203          ASR     R3           ; DIVIDE VECTOR BY 4 FOR UDA INITIALIZATION
21 030356 052703 104600   BIS     #104600,R3      ; SET OTHER BITS FOR UDA INITIALIZATION
22 030362 005037 002236   CLR     INTRCV        ; FLAG AS NO INTERRUPTS RECEIVED
23 030366 005014          CLR     (R4)         ; INIT UDA
24 030370 012737 004000 023652   MOV     #SA.S1,UDARSD  ; LOOK FOR STEP 1 COMPLETION
25 030376 004737 023514   CALL   UDARSP        ; WAIT FOR COMPLETION
26 030402 010364 000002   MOV     R3,2(R4)     ; MOVE STEP 1 DATA TO UDA
27 030406 012700 000012   MOV     #10.,R0      ; SET UP TIMEOUT OF 10 SECONDS
28 030412 010501          MOV     R5,R1
29 030414 062701 000040   ADD     #C.TC,R1     ; POINT TO CONTROLLER TABLE
30 030420 004737 022660   CALL   SETTO
31 030424 005737 002236          9$:  TST     INTRCV        ; SEE IF INTERRUPTED
32 030430 001022          BNE    11$          ; IF SO, EVERYTHING'S OK, SO BRANCH
33 030432          BREAK
    030432 104422          TRAP   C$BRK
34 030434 005737 002352          TST     KW.CSR       ; SEE IF CLOCK ON SYSTEM
35 030440 001771          BEQ    9$
36 030442 023765 002364 000042   CMP     KW.EL+2,C.TOH(R5) ; SEE IF TIME ELAPSED
37 030450 101005          BHI    3$
38 030452 001364          BNE    9$
39 030454 023765 002362 000040   CMP     KW.EL,C.TO(R5)
40 030462 103760          BLO    9$
41 030464          3$:  ERRDF  59,INTST4 ; REPORT NO INTERRUPTS ERROR

```

```

030464 104455
030466 000073
030470 005124
030472 000000
42 030474 000474
43 030476 005037 002236
44 030502 012700 000340
030506 104441
45 030510 005064 000002
46 030514 012702 000144
47 030520 005302
48 030522 001376
49 030524 012701 000007
50 030530
030530 010146
51 030532 012702 000005
52 030536 006301
53 030540 005302
54 030542 001375
55 030544
030544 010100
030546 104441
56 030550
030550 012601
57 030552 005737 002236
58 030556 001007
59 030560 005301
60 030562 100362
61 030564
030564 104455
030566 000073
030570 005124
030572 000000
62 030574 000434
63 030576
030576 012700 000000
030602 104441
64 030604 005201
65 030606 023701 030706
66 030612 001405
67 030614
030614 104455
030616 000074
030620 005101
030622 012566
68 030624 000420
69 030626
030626 012746 006771
030632 012746 000001
030636 010600
030640 104417
030642 062706 000004
70 030646 016503 000004
71 030652 042703 177000
72 030656
030656 010300

```

```

TRAP C$ERDF
.WORD 59
.WORD INTST4
.WORD 0
11$: BR 6$ : BRANCH
CLR INTRCV : FLAG AS NO INTERRUPTS RECEIVED
SETPRI #PRI07 : SET PRIORITY AS HIGHEST PRIORITY
MOV #PRI07,R0
TRAP C$SPRI
12$: CLR 2(R4) : WRITE SECOND STEP TO UDA
MOV #100.,R2 : SET UP DELAY SO WE KNOW WE'RE INTERRUPTED
DEC R2 : DECREMENT COUNT
BNE 12$ : IF INCOMPLETE, BRANCH
MOV #7.,R1 : R1 IS PROCESS PRIORITY LEVEL
PUSH R1 : SAVE PRIORITY
MOV R1,-(SP)
10$: MOV #5.,R2 : SET UP FOR SHIFTING PRIORITY
ASL R1 : SHIFT PRIORITY
DEC R2 : DECREMENT SHIFT COUNT
BNE 10$ : IF INCOMPLETE, BRANCH
SETPRI R1 : SET RUNNING PRIORITY TO R1
MOV R1,R0
TRAP C$SPRI
POP R1 : RESTORE R1
MOV (SP)+,R1
TST INTRCV : SEE IF INTERRUPT RECEIVED
BNE 4$ : IF SO, BRANCH
DEC R1 : DECREMENT PRIORITY LEVEL
BPL 2$ : IF ALL LEVELS UNTESTED, BRANCH
ERRDF 59,INTST4 : REPORT NO INTERRUPTS ERROR
TRAP C$ERDF
.WORD 59
.WORD INTST4
.WORD 0
4$: BR 6$ : BRANCH
SETPRI #PRI00 : SET RUNNING PRIORITY TO 0
MOV #PRI00,R0
TRAP C$SPRI
INC R1 : SO PRIORITY = BR LEVEL
CMP BRLEV,R1 : SEE IF BR LEVEL MATCHES PRIORITY
BEQ 5$ : IF SO, BRANCH
ERRDF 60,INTST2,INTERR : REPORT ERROR
TRAP C$ERDF
.WORD 60
.WORD INTST2
.WORD INTERR
5$: BR 6$ : BRANCH
PRINTF #INTST1 : PRINT TESTING COMPLETED
MOV #INTST1,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP
MOV C.VEC(R5),R3 : GET VECTOR ADDRESS
BIC #^CCT.VEC,R3 : CLEAR UNUSED BITS
CLRVEC R3 : CLEAR VECTOR
MOV R3,R0

```

030660 104436
 73 030662 000244
 74 030664 000207
 75 030666 016503 000004
 76 030672 042703 177000
 77 030676
 030676 010300
 030700 104436
 78 030702 000264
 79 030704 000207
 80
 81
 82
 83
 84 030706

6\$: CLZ
 RETURN
 MOV C.VEC(R5),R3
 BIC #^CCT.VEC,R3
 CLRVEC R3

 SEZ
 RETURN

 BRLEV: .BLKW 1

; FLAG AS NO ERRORS TRAP C\$CVEC
 ; RETURN TO TEST 1
 ; GET VECTOR ADDRESS
 ; CLEAR UNUSED BITS
 ; CLEAR VECTOR

 MOV R3,R0
 TRAP C\$CVEC
 ; FLAG AS ERROR OCCURRED
 ; RETURN TO TEST 1

 ; WORD FOR BRANCH LEVEL STORAGE

```

1          .SBTTL TEST 2: DISK RESIDENT DIAGNOSTIC TEST
2
3 030710          BGNTST
4          T2::
5 030710 012701 000002          MOV #2,R1          ;INIT TEST PARAMETERS
6 030714 004737 013006          CALL TINIT
7
8 030720 013737 002204 002210          MOV CTABS,TSTTAB          ;GET POINTER TO FIRST CONTROLLER TABLE
9
10 030726 004737 024414          T2NEXT: CALL RESET          ;RESET ALL UNITS
11 030732          PUSH FFREE          ;SAVE FREE MEMORY PARAMETERS
12 030732 013746 002174          MOV FFREE,-(SP)
13 030736 013746 002176          PUSH FSIZE
14 030742 012701 000001          MOV FSIZE,-(SP)
15 030746 004737 013134          MOV #1,R1          ;RUN DM PROGRAM IN
16 030752 001402          CALL RUNDM          ; ONE CONTROLLER ONLY
17 030754 004737 013224          BEQ 1$
18 030760 012637 002176          CALL RESPDM
19 030764 012637 002174          POP FSIZE
20 030770 062737 000046 002210          POP FSIZE          MOV (SP)+,FSIZE
21 030776 005777 151206          POP FFREE          MOV (SP)+,FFREE
22 031002 001351          ADD #C.SIZE,TSTTAB          ;MOVE TO NEXT CONTROLLER
23          TST @TSTTAB          ;CHECK IF ANY MORE CONTROLLER TABLES
24          BNE T2NEXT
          ENDTST
          L10041: TRAP C$ETST
031004 104401

```

```

1          .SBTTL TEST 3: DISK FUNCTION TEST
2
3 031006          BGNTST
4          T3::
5 031006 012701 000003          MOV #3,R1          ;INITIALIZE TEST PARAMETERS
6 031012 004737 013006          CALL TINIT
7
8 031016 013737 002204 002210          MOV CTABS,TSTTAB          ;GET FIRST TABLE ADDRESS
9 031024 013701 002206          MOV CTRLRS,R1          ;RUN DM PROGRAM ON ALL CONTROLLERS
10 031030 004737 013134          CALL RUNDM          ; AT ONCE
11 031034 001402          BEQ 1$
12 031036 004737 013224          CALL RESPDM
13 031042          1$:
14 031042          ENDTST
          031042          L10042: TRAP C$ETST
          031042 104401

```

```

1          .SBTTL TEST 4: DISK EXERCISER
2
3 031044   BGNTST
4          T4::
5 031044 022737 000004 002226   CMP #4,TNLM           ;CHECK IF TEST 4 WAS IN PROGRESS
6 031052 001066                   BNE T4STRT          ;BRANCH IF NOT
7 031054 022737 000002 002224   CMP #ICONT,IFLAGS   ;CHECK IF HERE BY CONTINUE COMMAND
8 031062 001066                   BNE T4STRT          ;BRANCH IF NOT
9 031064 005037 002224           CLR IFLAGS          ;CLEAR FLAGS FOR NEXT TIME HERE
10 031070 013704 002404          MOV LBUFS,R4        ;GET LOG BUFFER POINTER
11 031074 001433                   BEQ LOGCHK          ; IF ZERO, NONE EXISTS
12 031076                   PRINTF #LOGM1          ;INTRODUCE ERROR LOG
13 031076 012746 011640           MOV #LOGM1,-(SP)
14 031102 012746 000001           MOV #1,-(SP)
15 031106 010600                   MOV SP,R0
16 031110 104417                   TRAP C$PNTF
17 031112 062706 000004           ADD #4,SP
18 031116 005037 002404          CLR LBUFS          ;CLEAR START ADDRESS TO ERASE BUFFER
19 031122 012405                   LOGOUT: MOV (R4)+,R5 ;GET CONTROLLER TABLE ADDRESS
20 031124 004737 020746           CALL PNERR          ;PRINT ERROR REPORT
21 031130 062704 000072           ADD #<HC.BSZ-2>,R4 ;BUMP POINTER TO NEXT ENTRY
22 031134 020437 002406           CMP R4,LBUFN        ;CHECK IF AT END
23 031140 103770                   BLO LOGOUT          ;PRINT ALL ENTRIES
24 031142                   PRINTF #LOGM2
25 031142 012746 011675           MOV #LOGM2,-(SP)
26 031146 012746 000001           MOV #1,-(SP)
27 031152 010600                   MOV SP,R0
28 031154 104417                   TRAP C$PNTF
29 031156 062706 000004           ADD #4,SP
30 031162 000414                   BR T4CON
31 031164 032737 001000 002160   LOGCHK: BIT #SM.LOG,SFPTBL+SO.BIT ;CHECK IF LOG ENABLED
32 031172 001410                   BEQ T4CON
33 031174                   PRINTF #LOGM3          ;REPORT LOG EMPTY
34 031174 012746 011724           MOV #LOGM3,-(SP)
35 031200 012746 000001           MOV #1,-(SP)
36 031204 010600                   MOV SP,R0
37 031206 104417                   TRAP C$PNTF
38 031210 062706 000004           ADD #4,SP
39 031214 005737 002232          T4CON: TST URNING   ;CHECK IF ANY CONTROLLERS STILL RUNNING
40 031220 001403                   BEQ T4STRT          ;RESTART IF NOT
41 031222 004737 013224          CALL RESPDM         ;CONTINUE BY RESPONDING TO REQUESTS
42 031226 000572                   BR T4WAIT           ;END OF TEST WHEN DONE
43          ;START TEST 4
44 031230 032737 000014 002224   T4STRT: BIT #ISTRT+IREST,IFLAGS ;HERE FROM OPERATOR COMMAND?
45 031236 001543                   BEQ T4RUN           ;RUN WITH PREVIOUS PARAMETERS IF NEW PASS
46 031240 032737 000200 002160   BIT #SM.MAN,SFPTBL+SO.BIT ;MANUAL INTERVENTION MODE?
47 031246 001505                   BEQ T4DEF           ;IF NOT, SET UP DEFAULT PARAMETERS
48 031250                   MANUAL              ;MANUAL INTERVENTION ALLOWED?
49 031250 104450                   TRAP C$MANI
50 031252                   BNCOMPLETE T4DEFW   ;IF NOT, GIVE WARNING
51 031252 103073                   BCC T4DEFW
52          ;INPUT PARAMETERS

```

```

40
41 031254 005037 002234          CLR UCNT          ;CLEAR COUNT OF UNITS USING PATTERN 16
42 031260 013705 002204          MOV CTABS,R5     ;GET FIRST CONTROLLER TABLE
43 031264 012702 000010          T4PRM1: MOV #8,R2 ;GET COUNT OF DRIVE TABLES
44 031270 010504                MOV R5,R4        ;GET FIRST DRIVE TABLE POINTER
45 031272 062704 000020          ADD #C.DR0,R4
46 031276 012403                T4PRM2: MOV (R4)+,R3 ;GET DRIVE TABLE ADDRESS
47 031300 001416                BEQ T4PRM4       ;GO TO NEXT CONTROLLER IF NONE
48 031302 032763 100000 000002   BIT #DT.AVL,D.UNIT(R3) ;SEE IF TO BE TESTED
49 031310 001010                BNE T4PRM3
50 031312 004737 031636          CALL T4QUEST     ;ASK QUESTIONS
51 031316 022763 000020 000006   CMP #16,D.PAT(R3)
52 031324 001002                BNE T4PRM3
53 031326 005237 002234          INC UCNT
54 031332 005302                T4PRM3: DEC R2    ;COUNT DRIVE TABLES
55 031334 001360                BNE T4PRM2       ;GO LOOK AT NEXT
56 031336 062705 000046          T4PRM4: ADD #C.SIZE,R5 ;GO TO NEXT CONTROLLER
57 031342 005715                TST (R5)         ; IF THERE IS ONE
58 031344 001347                BNE T4PRM1

```

;NOW GET DATA PATTERN 16 IF SELECTED BY ANY DRIVE

```

60
61
62 031346 005737 002234          TST UCNT         ;CHECK IF ANY UNITS USING PAT 16
63 031352 001475                BEQ T4RUN        ;IF NONE, DON'T ASK ANY MORE QUESTIONS
64 031354                GMANID T4DPC,PAT16C,D,-1,1,16.,YES ;COUNT OF WORDS

```

```

TRAP  CS$GMAN
BR     10000$
.WORD  PAT16C
.WORD  T$CODE
.WORD  T4DPC
.WORD  -1
.WORD  T$LOLIM
.WORD  T$HILIM

```

10000\$:

```

65 031374 013701 002310          MOV PAT16C,R1    ;GET COUNT OF WORDS
66 031400 012704 002312          MOV #PAT16W,R4  ;GET ADDRESS OF STORAGE
67 031404 011437 002260          T4PRM5: MOV (R4),TEMP
68 031410                GMANID T4DPD,TEMP,0,-1,1,-1,YES ;DATA WORD

```

```

TRAP  CS$GMAN
BR     10001$
.WORD  TEMP
.WORD  T$CODE
.WORD  T4DPD
.WORD  -1
.WORD  T$LOLIM
.WORD  T$HILIM

```

10001\$:

```

69 031430 013724 002260          MOV TEMP,(R4)+
70 031434 005301                DEC R1           ;COUNT THE WORDS
71 031436 001362                BNE T4PRM5
72 031440 000442                BR T4RUN

```

;GIVE WARNING MANUAL INTERVENTION NOT ALLOWED

```

73
74
75
76 031442                T4DEFW: PRINTF #T4WARN
    031442 012746 007212          MOV #T4WARN,-(SP)
    031446 012746 000001          MOV #1,-(SP)

```



```

031452 010600
031454 104417
031456 062706 000004
77
78
79
80 031462 013705 002204
81 031466 012702 000010
82 031472 010504
83 031474 062704 000020
84 031500 012403
85 031502 001415
86 031504 062703 000004
87 031510
031510 042713 157777
88 031514 052723 011012
89 031520 012700 000067
90 031524 005023
91 031526 005300
92 031530 001375
93 031532 005302
94 031534 001361
95 031536 062705 000046
96 031542 005715
97 031544 001350
98
99
100
101 031546 006137 002224
102 031552
031552 042737 177757 002224
103 031560 012701 000004
104 031564 004737 013006
105
106 031570 013737 002204 002210
107 031576 013701 002206
108 031602 004737 013134
109 031606 001402
110 031610 004737 013224
111 031614 032737 001000 002160
112 031622 001402
113 031624
031624 104422
114 031626 000772
115
116 031630
031630 104424
117 031632
031632 104432
031634 001732

```

```

;SET UP DEFAULT PARAMETERS
T4DEF: MOV CTABS,R5 ;GET FIRST CONTROLLER TABLE
T4DEFA: MOV #8.,R2 ;GET COUNT OF DRIVE TABLES
MOV R5,R4 ;GET FIRST DRIVE TABLE POINTER
ADD #C.DRO,R4
T4DEFB: MOV (R4)+,R3 ;GET DRIVE TABLE ADDRESS
BEQ T4DEFE ;GO TO NEXT CONTROLLER IF NONE
ADD #D.PRM,R3
AND D.DCY,(R3) ;INITIALIZE ALL PARAMETER BITS
BIC #^C<D.DCY>,(R3)
BIS #DDEF,(R3)+
MOV #55.,R0
T4DEFC: CLR (R3)+
DEC R0
BNE T4DEFC
T4DEFD: DEC R2 ;COUNT DRIVE TABLES
BNE T4DEFB ;GO LOOK AT NEXT
T4DEFE: ADD #C.SIZE,R5 ;GO TO NEXT CONTROLLER
TST (R5) ; IF THERE IS ONE
BNE T4DEFA

;START TEST 4
T4RUN: ROL IFLAGS ;CLEAR FLAGS FOR NEXT TIME HERE
AND ISTRTH,IFLAGS ;HOLD START FOR T4UPRM REQUEST
BIC #^C<ISTRTH>,IFLAGS
MOV #4,R1 ;INITIALIZE TEST PARAMETERS
CALL TINIT

MOV CTABS,TSTTAB ;GET FIRST TABLE ADDRESS
MOV CTRLRS,R1 ;RUN DM PROGRAM ON ALL CONTROLLERS
CALL RUNDM ; AT ONCE
BEQ T4WAIT
CALL RESPDM
T4WAIT: BIT #SM.LOG,SFPTBL+SO.BIT ;CHECK IF LOG IS ENABLED
BEQ T4EXIT ;EXIT IF NOT
BREAK

BR T4WAIT ;WAIT TILL STOPPED BY CONTROL C

T4EXIT: DORPT ;PRINT STATISTICS
EXIT TST

```

```

MOV SP,R0
TRAP C$PNTF
ADD #4,SP

TRAP C$BRK

TRAP C$DRPT

TRAP C$EXIT
.WORD L10043-

```

```

1      :ASK TEST 4 MANUAL INTERVENTION QUESTIONS
2
3      :INPUTS:
4      R5 - POINTER TO CONTROLLER TABLE
5      R3 - POINTER TO DRIVE TABLE
6      R2 AND R4 MUST BE PRESERVED
7
8      :OUTPUTS:
9      DRIVE TABLE WITH NEW PARAMETERS
10     R0 AND R1 CONTENTS DESTROYED
11     031636      T4QUEST:PUSH <R2,R4>
12     031636      MOV R2,-(SP)
13     031640      MOV R4,-(SP)
14     031642      PRINTF #T4QHED,D.UNIT(R3),(R5),(R3) ;PRINT HEADER
15     031642      MOV (R3),-(SP)
16     031644      MOV (R5),-(SP)
17     031646      MOV D.UNIT(R3),-(SP)
18     031652      MOV #T4QHED,-(SP)
19     031656      MOV #4,-(SP)
20     031662      MOV SP,R0
21     031664      TRAP C$PNTF
22     031666      ADD #12,SP
23     031672      MOV D.BB(R3),TEMP
24     031700      GMANID T4BB,TEMP,D,-1,0,16.,YES ;NUMBER OF BAD BLOCKS
25     031700      TRAP C$GMAN
26     031702      BR 10002$
27     031704      .WORD TEMP
28     031706      .WORD T$CODE
29     031710      .WORD T4BB
30     031712      .WORD -1
31     031714      .WORD T$LLOLIM
32     031716      .WORD T$HILIM
33     031720      10002$:
34     031720      MOV TEMP,D.BB(R3)
35     031726      BEQ T4Q02
36
37     031730      MOV R3,R4 ;GET POINTER TO STORAGE
38     031732      ADD #D.BB01,R4 ;FOR BAD BLOCKS
39     031736      MOV TEMP,R1 ;GET COUNT OF BLOCKS TO INPUT
40
41     031742      T4Q01: CALL BLD28 ;BUILD DEFAULT ANSWER
42     031746      GMANID T4BBI,TEMP,A,-1,0,9.,YES ;BAD BLOCK
43     031746      TRAP C$GMAN
44     031750      BR 10003$
45     031752      .WORD TEMP
46     031754      .WORD T$CODE
47     031756      .WORD T4BBI
48     031760      .WORD -1
49     031762      .WORD T$LLOLIM
50     031764      .WORD T$HILIM
51     031766      10003$:
52     031766      CALL CNV28 ;CONVERT TO BINARY
53     031772      BCS T4Q01 ;REPEAT UNTIL RIGHT
54     031774      DEC R1 ;DECREMENT COUNT
55     031776      BNE T4Q01 ;GET ALL NUMBERS
56     032000      MOV D.PRM(R3),TEMP ;GET PARAMETER BITS
57     032006      GMANIL T4RO,TEMP,D.RO,YES ;READ ONLY

```

```

032006 104443 TRAP C$GMAN
032010 000404 BR 10004$
032012 002260 .WORD TEMP
032014 000130 .WORD T$CODE
032016 003632 .WORD T4RO
032020 004000 .WORD D.RO
032022 10004$:
30 032022 032737 004000 002260 BIT #D.RO,TEMP
31 032030 001056 BNE T4Q06
32 032032 T4Q03: GMANIL T4WO,TEMP,D.WO,YES ;WRITE ONLY
032032 104443 TRAP C$GMAN
032034 000404 BR 10005$
032036 002260 .WORD TEMP
032040 000130 .WORD T$CODE
032042 003644 .WORD T4WO
032044 002000 .WORD D.WO
032046 10005$:
33 032046 GMANIL T4WCA,TEMP,D.WCA,YES ;CHECK ALL WRITES
032046 104443 TRAP C$GMAN
032050 000404 BR 10006$
032052 002260 .WORD TEMP
032054 000130 .WORD T$CODE
032056 003657 .WORD T4WCA
032060 000004 .WORD D.WCA
032062 10006$:
34 032062 032737 000004 002260 BIT #D.WCA,TEMP ;CHECK ANSWER
35 032070 001007 BNE T4Q04 ;BRANCH IF YES
36 032072 GMANIL T4WCR,TEMP,D.WC,YES ;RANDOMLY CHECK WRITES
032072 104443 TRAP C$GMAN
032074 000404 BR 10007$
032076 002260 .WORD TEMP
032100 000130 .WORD T$CODE
032102 003713 .WORD T4WCR
032104 000010 .WORD D.WC
032106 10007$:
37 032106 000403 BR T4Q05
38 032110 052737 000010 002260 T4Q04: BIS #D.WC,TEMP ;BOTH BITS GET SET
39 032116 013763 002260 000004 T4Q05: MOV TEMP,D.PRM(R3) ;PUT PARAM BITS BACK
40 032124 016337 000006 002260 MOV D.PAT(R3),TEMP
41 032132 GMANID T4DP,TEMP,D,-1,0,16.,YES ;DATA PATTERN
032132 104443 TRAP C$GMAN
032134 000406 BR 10010$
032136 002260 .WORD TEMP
032140 000052 .WORD T$CODE
032142 003754 .WORD T4DP
032144 177777 .WORD -1
032146 000000 .WORD T$LOLIM
032150 000020 .WORD T$HILIM
032152 10010$:
42 032152 013763 002260 000006 MOV TEMP,D.PAT(R3)
43 032160 016337 000004 002260 MOV D.PRM(R3),TEMP
44 032166 032737 004000 002260 T4Q06: BIT #D.RO,TEMP ;GET PARAM BITS AGAIN
45 032174 001010 BNE T4Q07 ;BYPASS NEXT 3 IF ONLY WRITING
46 032176 032737 002000 002260 BIT #D.WO,TEMP
47 032204 001404 BEQ T4Q07
48 032206 032737 000010 002260 BIT #D.WC,TEMP
49 032214 001432 BEQ T4Q09

```

```

50 032216          T4Q07:  GMANIL T4ECC,TEMP,D.ECC,YES      ;ENABLE ECC
    032216 104443
    032220 000404
    032222 002260
    032224 000130
    032226 004022
    032230 010000
    032232
    51 032232          GMANIL T4DCA,TEMP,D.DCA,YES      ;COMPARE ALL DATA
    032232 104443
    032234 000404
    032236 002260
    032240 000130
    032242 004055
    032244 000001
    032246
    52 032246 032737 000001 002260      BIT #D.DCA,TEMP      ;CHECK ANSWER
    53 032254 001007
    54 032256          GMANIL T4DCR,TEMP,D.DC,YES      ;RANDOMLY CHECK WRITES
    032256 104443
    032260 000404
    032262 002260
    032264 000130
    032266 004103
    032270 000002
    032272
    55 032272 000403
    56 032274 052737 000002 002260      T4Q08:  BR T4Q09
    57 032302          T4Q09:  GMANIL T4RET,TEMP,D.RET,YES  ;BOTH BITS GET SET
    032302 104443
    032304 000404
    032306 002260
    032310 000130
    032312 004136
    032314 001000
    032316
    58 032316 005137 002260
    59 032322          COM TEMP
    032322 104443          GMANIL T4SEK,TEMP,D.SEQ,YES      ;ENABLE SEEKS
    032324 000404
    032326 002260
    032330 000130
    032332 004155
    032334 000100
    032336
    60 032336 005137 002260
    61 032342 013763 002260 000004      COM TEMP
    62
    63 032350 005037 002260
    64 032354 032763 000040 000004      MOV TEMP,D.PRM(R3)
    65 032362 001403
    66 032364 005237 002260
    67 032370 000422
    68 032372 032763 000400 000004      CLR TEMP
    69 032400 001416
    70 032402 012737 000004 002260      BIT #D.BE,D.PRM(R3)
    71 032410 005763 000112
    MOV #4,TEMP
    TST D.BEC(R3)

```

10011\$:

10012\$:

10012\$:

10013\$:

10014\$:

10015\$:

TRAP CS\$GMAN
BR 10011\$
.WORD TEMP
.WORD T\$CODE
.WORD T4ECC
.WORD D.ECC

TRAP CS\$GMAN
BR 10012\$
.WORD TEMP
.WORD T\$CODE
.WORD T4DCA
.WORD D.DCA

TRAP CS\$GMAN
BR 10013\$
.WORD TEMP
.WORD T\$CODE
.WORD T4DCR
.WORD D.DC

TRAP CS\$GMAN
BR 10014\$
.WORD TEMP
.WORD T\$CODE
.WORD T4RET
.WORD D.RET

TRAP CS\$GMAN
BR 10015\$
.WORD TEMP
.WORD T\$CODE
.WORD T4SEK
.WORD D.SEQ

;COMPLIMENTED

;DETERMINE DEFAULT SELECTION
;IF D.BE SET - LOAD 1
;IF D.CYL CLEAR - LOAD 0
;IF D.BEC CONTAINS 0 - LOAD 4
;IF D.TR SET - LOAD 2
;LOAD 3

TEST 4: DISK EXERCISER

```

72 032414 001410          BEQ T4Q11
73 032416 005337 002260   DEC TEMP
74 032422 032763 000020 000004   BIT #D.TR,D.PRM(R3)
75 032430 001402          BEQ T4Q11
76 032432 005337 002260   DEC TEMP
77 032436          T4Q11: PRINTF #T4OPT1
    032436 012746 007432          MOV #T4OPT1,-(SP)
    032442 012746 000001          MOV #1,-(SP)
    032446 010600          MOV SP,R0
    032450 104417          TRAP C$PNTF
    032452 062706 000004          ADD #4,SP
78 032456          PRINTF #T4OPT2
    032456 012746 007460          MOV #T4OPT2,-(SP)
    032462 012746 000001          MOV #1,-(SP)
    032466 010600          MOV SP,R0
    032470 104417          TRAP C$PNTF
    032472 062706 000004          ADD #4,SP
79 032476          PRINTF #T4OPT3
    032476 012746 007525          MOV #T4OPT3,-(SP)
    032502 012746 000001          MOV #1,-(SP)
    032506 010600          MOV SP,R0
    032510 104417          TRAP C$PNTF
    032512 062706 000004          ADD #4,SP
80 032516          PRINTF #T4OPT4
    032516 012746 007577          MOV #T4OPT4,-(SP)
    032522 012746 000001          MOV #1,-(SP)
    032526 010600          MOV SP,R0
    032530 104417          TRAP C$PNTF
    032532 062706 000004          ADD #4,SP
81 032536          PRINTF #T4OPT5
    032536 012746 007657          MOV #T4OPT5,-(SP)
    032542 012746 000001          MOV #1,-(SP)
    032546 010600          MOV SP,R0
    032550 104417          TRAP C$PNTF
    032552 062706 000004          ADD #4,SP
82 032556          PRINTF #T4OPT6
    032556 012746 007737          MOV #T4OPT6,-(SP)
    032562 012746 000001          MOV #1,-(SP)
    032566 010600          MOV SP,R0
    032570 104417          TRAP C$PNTF
    032572 062706 000004          ADD #4,SP
83 032576          GMANID T4OPT7,TEMP,D,-1,0,4,YES ;WHICH SELECTION LIMITS
    032576 104443          TRAP C$GMAN
    032600 000406          BR 10016$
    032602 002260          .WORD TEMP
    032604 000052          .WORD T$CODE
    032606 004176          .WORD T4OPT7
    032610 177777          .WORD -1
    032612 000000          .WORD T$LOLIM
    032614 000004          .WORD T$HILIM
    032616          10016$:
84 032616 005337 002260   DEC TEMP          ;SET UP D.PRM FROM ANSWER
85 032622 002004          BGE T4Q12          ;IF 0 - CLEAR D.BE AND D.CYL
86 032624 042763 000440 000004   BIC #D.BE+D.CYL,D.PRM(R3)
87 032632 000467          BR T4Q19
88 032634 005337 002260   DEC TEMP          ;IF 1
89 032640 002013          BGE T4Q13          ; IF D.BE NOT SET

```

```

90 032642 032763 000040 000004 BIT #D.BE,D.PRM(R3) ; SET D.BE
91 032650 001060 BNE T4Q19 ; CLEAR D.CYL
92 032652 052763 000040 000004 BIS #D.BE,D.PRM(R3) ; LOAD 1 IN D.BEC
93 032660 042763 000400 000004 BIC #D.CYL,D.PRM(R3) ; CLEAR BLOCK STORAGE
94 032666 000436 BR T4Q16
95 032670 042763 000040 000004 T4Q13: BIC #D.BE,D.PRM(R3) ;IF 2, 3 OR 4
96 ; CLEAR D.BE
97 032676 022737 000002 002260 CMP #2,TEMP ;IF 4
98 032704 001006 BNE T4Q14 ; SET D.CYL
99 032706 052763 000400 000004 BIS #D.CYL,D.PRM(R3) ; CLEAR D.BEC
100 032714 005063 000112 CLR D.BEC(R3)
101 032720 000434 BR T4Q19
102 032722 T4Q14: PUSH D.PRM(R3) ;IF 2 OR 3
032722 016346 000004 MOV D.PRM(R3),-(SP)
103 032726 052763 000420 000004 BIS #D.CYL+D.TR,D.PRM(R3) ; SAVE D.PRM BITS
104 032734 005337 002260 DEC TEMP ; SET D.CYL AND D.TR
105 032740 100403 BMI T4Q15 ; IF 3
106 032742 042763 000020 000004 T4Q15: BIC #D.TR,D.PRM(R3) ; CLEAR D.TR
107 032750 022663 000004 CMP (SP)+,D.PRM(R3) ; IF D.CYL OR D.TR CHANGED OR D.BEC = 0
108 032754 001003 BNE T4Q16
109 032756 005763 000112 TST D.BEC(R3) ; LOAD 1 IN D.BEC
110 032762 001013 BNE T4Q19 ; CLEAR BLOCK STORAGE
111 032764 012763 000001 000112 T4Q16: MOV #1,D.BEC(R3)
112 032772 010304 T4Q17: MOV R3,R4
113 032774 062704 000114 ADD #D.BGN1,R4
114 033000 012701 000020 MOV #16.,R1
115 033004 005024 T4Q18: CLR (R4)+
116 033006 005301 DEC R1
117 033010 001375 BNE T4Q18
118 033012 032763 000040 000004 T4Q19: BIT #D.BE,D.PRM(R3) ;NOW ASK THE QUESTIONS TO ALLOW THE
119 033020 001460 BEQ T4Q22 ; NUMBERS TO CHANGE
120 033022 016337 000112 002260 MOV D.BEC(R3),TEMP ;NUMBER OF B/E SETS
121 033030 GMANID T4BE,TEMP,D,-1,1,4,YES
033030 104443 TRAP C$GMAN
033032 000406 BR 10017$
033034 002260 .WORD TEMP
033036 000052 .WORD T$CODE
033040 004201 .WORD T4BE
033042 177777 .WORD -1
033044 000001 .WORD T$LOLIM
033046 000004 .WORD T$HILIM
033050 10017$:
122 033050 013763 002260 000112 MOV TEMP,D.BEC(R3)
123 033056 013701 002260 MOV TEMP,R1 ;GET COUNT OF SETS
124 033062 010304 MOV R3,R4 ;GET POINTER TO STORAGE AREA
125 033064 062704 000114 ADD #D.BGN1,R4
126 033070 004737 020126 T4Q20: CALL BLD28
127 033074 GMANID T4BEG,TEMP,A,-1,0,9.,YES ;BEGIN BLOCK
033074 104443 TRAP C$GMAN
033076 000406 BR 10020$
033100 002260 .WORD TEMP
033102 000152 .WORD T$CODE
033104 004232 .WORD T4BEG
033106 177777 .WORD -1
033110 000000 .WORD T$LOLIM
033112 000011 .WORD T$HILIM
033114 10020$:

```

TEST 4: DISK EXERCISER

```

128 033114 004737 020230          CALL CNV28
129 033120 103763          BCS T4Q20
130 033122 004737 020126      T4Q21: CALL BLD28
131 033126          GMANID T4END,TEMP,A,-1,0,9.,YES ;END BLOCK
    033126 104443          TRAP      CS$GMAN
    033130 000406          BR        10021$
    033132 002260          .WORD    TEMP
    033134 000152          .WORD    T$CODE
    033136 004246          .WORD    T4END
    033140 177777          .WORD    -1
    033142 000000          .WORD    T$LOLIM
    033144 000011          .WORD    T$HILIM
    033146          10021$:

132 033146 004737 020230          CALL CNV28
133 033152 103763          BCS T4Q21
134 033154 005301          DEC R1
135 033156 001344          BNE T4Q20
136 033160 000577          BR T4Q30
137 033162 032763 000400 000004 T4Q22: BIT #D.CYL,D.PRM(R3)          ;IF D.CYL CLEAR - ALL DONE
138 033170 001573          BEQ T4Q30
139 033172 005763 000112          TST D.BEC(R3)          ;IF D.BEC CLEAR - GO RIGHT TO B/E CYLS
140 033176 001526          BEQ T4Q27
141 033200 010304          MOV R3,R4
142 033202 062704 000112          ADD #D.BEC,R4
143 033206 032763 000020 000004          BIT #D.TR,D.PRM(R3)          ;LOOK AT D.TR.TO DETERMINE QUESTION
144 033214 001434          BEQ T4Q24
145 033216 011437 002260          MOV (R4),TEMP
146 033222          GMANID T4TRC,TEMP,D,-1,1,7,YES ;NUMBER OF TRACKS
    033222 104443          TRAP      CS$GMAN
    033224 000406          BR        10022$
    033226 002260          .WORD    TEMP
    033230 000052          .WORD    T$CODE
    033232 004260          .WORD    T4TRC
    033234 177777          .WORD    -1
    033236 000001          .WORD    T$LOLIM
    033240 000007          .WORD    T$HILIM
    033242          10022$:

147 033242 013714 002260          MOV TEMP,(R4)
148 033246 012401          MOV (R4)+,R1          ;GET COUNT OF TRACKS
149 033250 011437 002260      T4Q23: MOV (R4),TEMP
150 033254          GMANID T4TRAK,TEMP,D,-1,0,255.,YES ;TRACK
    033254 104443          TRAP      CS$GMAN
    033256 000406          BR        10023$
    033260 002260          .WORD    TEMP
    033262 000052          .WORD    T$CODE
    033264 004311          .WORD    T4TRAK
    033266 177777          .WORD    -1
    033270 000000          .WORD    T$LOLIM
    033272 000377          .WORD    T$HILIM
    033274          10023$:

151 033274 013724 002260          MOV TEMP,(R4)+
152 033300 005301          DEC R1
153 033302 001362          BNE T4Q23
154 033304 000433          BR T4Q26
155 033306 011437 002260      T4Q24: MOV (R4),TEMP
156 033312          GMANID T4GRC,TEMP,D,-1,1,7,YES ;NUMBER OF GROUPS
    033312 104443          TRAP      CS$GMAN

```


181 033520 004737 020230
 182 033524 103763
 183 033526 004737 020126
 184 033532
 033532 104443
 033534 000406
 033536 002260
 033540 000152
 033542 004452
 033544 177777
 033546 000000
 033550 000011
 033552
 185 033552 004737 020230
 186 033556 103763
 187 033560
 033560 012604
 033562 012602
 188 033564 000207
 189 033566
 033566
 033566 104401
 190 033570
 191

T4Q29: CALL CNV28
 BCS T4Q28
 CALL BLD28
 GMANID T4CYCLE,TEMP,A,-1,0,9.,YES

;ENDING CYLINDER

TRAP
 BR 10030\$
 .WORD TEMP
 .WORD T\$CODE
 .WORD T4CYCLE
 .WORD -1
 .WORD T\$LOLIM
 .WORD T\$HILIM

10030\$:

T4Q30: CALL CNV28
 BCS T4Q29
 POP <R4,R2>
 MOV (SP)+,R4
 MOV (SP)+,R2

RETURN
ENDTST

L10043:

TRAP C\$ETST

ENDMOD

.SBTTL

1
9
10
38
39
40
41
42
43
44
45
46
47
48
49
50
51
61
62
63
64
65
66
67
68
69
70

033570

033570 000032
033570
033572

;.TITLE PARAMETER CODING (6)

.SBTTL HARDWARE PARAMETER CODING SECTION

BGNMOD

;++
; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
; WITH THE OPERATOR.
;--

BGNHRD

.WORD L10044-L\$HARD/2
L\$HARD::

;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:

HO.UBA ==	0.	; UNIBUS ADDRESS
HO.VEC ==	HO.UBA+2	; UDA VECTOR
HO.BRL ==	HO.VEC+2	; BR LEVEL
HO.BST ==	HO.BRL+2	; BURST RATE
HO.LDR ==	HO.BST+2	; DRIVE NUMBER
HO.PRM ==	HO.LDR+2	; PROGRAM PARAMETERS
HM.CYL ==	BIT13	; TEST CUSTOMER DATA AREA

```

1 033572          GPRMA  H.UBA,HO.UBA,0,160000,177774,YES      ;BUS ADDRESS
   033572 000031          .WORD  T$CODE
   033574 033656          .WORD  H.UBA
   033576 160000          .WORD  T$LOLIM
   033600 177774          .WORD  T$HILIM
2 033602          GPRMA  H.VEC,HO.VEC,0,4,774,YES      ; VECTOR
   033602 001031          .WORD  T$CODE
   033604 033704          .WORD  H.VEC
   033606 000004          .WORD  T$LOLIM
   033610 000774          .WORD  T$HILIM
3 033612          GPRMD  H.BRL,HO.BRL,D,-1,4.,7.,YES      ; BR LEVEL
   033612 002052          .WORD  T$CODE
   033614 033713          .WORD  H.BRL
   033616 177777          .WORD  -1
   033620 000004          .WORD  T$LOLIM
   033622 000007          .WORD  T$HILIM
4 033624          GPRMD  H.BST,HO.BST,D,-1,0.,63.,YES      ; BURST RATE
   033624 003052          .WORD  T$CODE
   033626 033724          .WORD  H.BST
   033630 177777          .WORD  -1
   033632 000000          .WORD  T$LOLIM
   033634 000077          .WORD  T$HILIM
5 033636          GPRMD  H.LDR,HO.LDR,D,-1,0.,255.,YES      ; DRIVE SELECT NUMBER
   033636 004052          .WORD  T$CODE
   033640 033746          .WORD  H.LDR
   033642 177777          .WORD  -1
   033644 000000          .WORD  T$LOLIM
   033646 000377          .WORD  T$HILIM
6 033650          GPRML  H.CST,HO.PRM,HM.CYL,YES ; USE CUSTOMER DATA AREA
   033650 005130          .WORD  T$CODE
   033652 033763          .WORD  H.CST
   033654 020000          .WORD  HM.CYL
7 033656          ENDHRD
                                     .EVEN
                                     L10044:

```

```

8
15 033656      125      116      111  H.UBA:  .ASCIZ  \UNIBUS ADDRESS OF UDA\
16 033704      126      105      103  H.VEC:  .ASCIZ  \VECTOR\
17 033713      102      122      040  H.BRL:  .ASCIZ  \BR LEVEL\
18 033724      125      116      111  H.BST:  .ASCIZ  \UNIBUS BURST RATE\
19 033746      104      122      111  H.LDR:  .ASCIZ  \DRIVE NUMBER\
20 033763      105      130      105  H.CST:  .ASCIZ  \EXERCISE ON CUSTOMER DATA AREA IN TEST 4\
21                                     .EVEN

```

1
2
3
4
5
6
7
8
9
10
11
12
13
22
23
24
25
26
27
28
29
30

```
.SBTTL SOFTWARE PARAMETER CODING SECTION
:++
: THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
: THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: WITH THE OPERATOR.
:--
```

```
034034
034034 000030
034036
```

BGNSFT

.WORD L10045-L\$SOFT/2
L\$SOFT::

:FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:

```
SO.EL= 0. ;ERROR LIMIT
SO.XL= 2. ;DATA TRANSFER LIMIT (MEGABITS)
SO.BIT= 4. ;SINGLE BIT ANSWERS
SM.MAN==BIT07 ; MANUAL INTERVENTION MODE
SM.SSF==BIT08 ; SUPPRESS SOFT ERRORS
SM.LOG==BIT09 ; ERROR LOG ENABLED
SM.IW== BIT14 ; INITIAL WRITE
```

```

1 034036          GPRML S.MAN,SO.BIT,SM.MAN,YES ;MANUAL INTERVENTION MODE
  034036 002130                                     .WORD T$CODE
  034040 034116                                     .WORD S.MAN
  034042 000200                                     .WORD SM.MAN
2 034044          DISPLAY S.MES ;MESSAGE ON NEXT QUESTIONS
  034044 000003                                     .WORD T$CODE
  034046 034203                                     .WORD S.MES
3 034050          GPRMD S.EL,SO.EL,D,-1,1.,-1.,YES ;ERROR LIMIT
  034050 000052                                     .WORD T$CODE
  034052 034266                                     .WORD S.EL
  034054 177777                                     .WORD -1
  034056 000001                                     .WORD T$LOLIM
  034060 177777                                     .WORD T$HILIM
4 034062          GPRMD S.XL,SO.XL,D,-1,0.,-1.,YES ;TRANSFER LIMIT
  034062 001052                                     .WORD T$CODE
  034064 034302                                     .WORD S.XL
  034066 177777                                     .WORD -1
  034070 000000                                     .WORD T$LOLIM
  034072 177777                                     .WORD T$HILIM
5 034074          GPRML S.SSF,SO.BIT,SM.SSF,YES ;SUPPRESS SOFT ERRORS
  034074 002130                                     .WORD T$CODE
  034076 034364                                     .WORD S.SSF
  034100 000400                                     .WORD SM.SSF
6 034102          GPRML S.IW,SO.BIT,SM.IW,YES ;INITIAL WRITE
  034102 002130                                     .WORD T$CODE
  034104 034422                                     .WORD S.IW
  034106 040000                                     .WORD SM.IW
7 034110          GPRML S.LOG,SO.BIT,SM.LOG,YES ;ERROR LOG
  034110 002130                                     .WORD T$CODE
  034112 034454                                     .WORD S.LOG
  034114 001000                                     .WORD SM.LOG
11 034116          ENDSFT
                                     .EVEN
                                     L10045:

```

```

12
13
20 034116      105      116      124  S.MAN: .ASCIZ\ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS\
21 034203      122      105      115  S.MES: .ASCIZ\REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY\
22 034265      000                                     .BYTE 0
23 034266      105      122      122  S.EL:  .ASCIZ\ERROR LIMIT\
24 034302      122      105      101  S.XL:  .ASCIZ\READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT\
25 034364      123      125      120  S.SSF: .ASCIZ\SUPPRESS PRINTING SOFT ERRORS\
26 034422      104      117      040  S.IW:  .ASCIZ\DO INITIAL WRITE ON START\
27 034454      105      116      101  S.LOG: .ASCIZ\ENABLE ERROR LOG\
31                                     .EVEN

```

1
2
3 034476
4 000050
5
6
13
14 034616
034616 034642
034620 000010
034622
15 034622
16

.SBTTL PATCH AREA
\$PATCH::
.REPT 40.
.WORD 0
.ENDR
LASTAD
L\$LAST::
ENDMOD
.SBTTL

.EVEN
.WORD T\$FREE
.WORD T\$SIZE

13 034622
14 034622
034622 000000
034624 000006
034626
15 034626 172150
16 034630 000154
17 034632 000005
18 034634 000000
19 034636 000000
20 034640 000000
21 034642
034642
22 034642

BGNSETUP 1
BGNPTAB

.WORD 172150
.WORD 154
.WORD 5.
.WORD 0
.WORD 0.
.WORD 0
ENDPTAB
ENDSETUP

.WORD 0
.WORD L10050-./2-1
L10046:
: UNIBUS ADDRESS
: VECTOR ADDRESS
: BR LEVEL
: UNIBUS BURST RATE
: LOGICAL DRIVE NUMBER
: CUSTOMER DATA AREA
L10050:

23
24
25 000001
ERRORS DETECTED: 0

.NLIST SYM
.END

VIRTUAL MEMORY USED: 29184 WORDS (114 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
.B:ZUDCAO/C=[20,0]SVC34R.MLB/P:1,B:ZUDCAO.DOC,B:ZUDCAO

DLLNAM	23-34#	40-41*	40-42*	50-31	93-47	93-49															
DLLR	23-31#	40-40*																			
DLLSIZ	23-33#	40-49*	40-54	40-55*	40-56*																
DLLV	23-30#	40-39*																			
DMEND	22-33#	31-20	32-16*																		
DMENDS	22-34#	31-21	32-17*																		
DMFRST	20-37#	93-64	93-74																		
DMMAIN	20-36#	69-5	69-7	69-30																	
DMNF	25-46#	32-20																			
DMOVRL	20-35#																				
DMPROG	22-32#	32-13*	53-25	53-26	57-7	57-8	67-42	67-43	68-42	68-45	68-47	68-48	69-4	69-6							
DMTRLN	20-34#																				
DONE	36-60	54-13#																			
DSPSIZ	36-4	36-62#																			
DT.AVL	19-10#	50-23	99-87	101-104	101-128	119-48															
DT.UNT	19-9#																				
DUP	11-34#	72-20																			
E\$END	5-16#																				
E\$LOAD	5-16#	5-77																			
EF.BBR	14-27#																				
EF.BBU	14-28#																				
EF.CON	9-70#	101-63																			
EF.LOG	14-29#																				
EF.NEW	9-70#																				
EF.PWR	9-70#	101-69																			
EF.RES	9-70#	101-66																			
EF.SEX	14-30#																				
EF.STA	9-70#	101-91																			
ENG	5-47#	5-74	8-24	22-50	34-28	68-15	68-49	75-16	75-18	75-23	123-31	124-8	124-28								
ERRBLK	22-21#																				
ERRC	55-13	55-20	56-103#																		
ERRCHR	22-23#	41-52*	41-53	41-58*	41-59	56-3*	56-4	56-6	56-13*	56-14	56-63*	56-64	61-58*	61-59							
	61-68*	61-69	65-3*	65-4	65-12*	65-13	65-29*	65-30													
ERRD	55-22	56-115#																			
ERRLIM	26-40#	52-44																			
ERRMB	26-35#	67-34																			
ERRMBD	26-36#	67-37																			
ERRMC	36-58	52-28#																			
ERRME1	26-38#	55-18	56-56																		
ERRMES	36-57	51-25#	52-29																		
ERRMRT	26-37#	98-23																			
ERRMSG	22-21#																				
ERRMSL	51-39	51-43#																			
ERRMSX	51-34	51-41	51-61	51-64#																	
ERRNBR	22-21#																				
ERRNL	26-21#	41-62	56-43	58-13	58-20	98-24	99-71														
ERRONE	26-20#	41-53	41-59	56-6	56-14	56-64	61-59	61-69	65-4	65-13	65-30										
ERRRSZ	56-51	56-73#																			
ERRRTB	56-54	56-71#	56-73																		
ERRRTN	67-26	67-31#																			
ERRTYP	22-21#	67-20																			
EVL	9-70#																				
F\$AU	5-16#																				
F\$AUTO	5-16#	110-10	110-18																		
F\$BGN	5-16#	5-43	8-29	9-48	27-14	27-19	27-24	27-30	27-34	27-39	27-51	27-56	27-61	27-65							
	27-69	27-73	27-77	27-81	67-31	81-10	82-18	95-5	96-1	98-28	99-38	99-45	100-8	101-8							
	108-45	110-10	111-8	111-31	112-16	112-18	112-28	112-28	112-40	112-41	112-41	112-43	112-44	112-44							

KW.VEC	23-5#	102-22*	102-24				
KW11I	95-5#	102-24					
KWOUT.	95-8	97-14	102-10#	102-25			
KYES	102-14	102-16	102-20#				
L\$ACP	5-77#						
L\$APT	5-77#						
L\$AUT	5-77#						
L\$AUTO	5-77	110-10#					
L\$CCP	5-77#						
L\$CLEA	5-77	111-8#					
L\$CO	5-77#						
L\$DEPO	5-77#						
L\$DESC	5-77	24-22#					
L\$DESP	5-77#						
L\$DEVP	5-77#						
L\$DISP	5-77	6-8#					
L\$DLY	5-77#	101-87					
L\$DTP	5-77#						
L\$DTYP	5-77#						
L\$DUT	5-77#						
L\$DVTY	5-77	24-12#					
L\$EF	5-77#						
L\$ENVI	5-77#						
L\$ERRT	5-77	22-21#					
L\$ETP	5-77#						
L\$EXP1	5-77#						
L\$EXP4	5-77#						
L\$EXP5	5-77#						
L\$HARD	5-77	121-50	121-50#				
L\$HIME	5-77#	39-51					
L\$HPCP	5-77#						
L\$HPTP	5-77#						
L\$HW	5-77	7-10	7-10#				
L\$ICP	5-77#						
L\$INIT	5-77	101-8#					
L\$LADP	5-77#						
L\$LAST	5-77	125-14#	126-22				
L\$LOAD	5-77#						
L\$LUN	5-77#	33-23*	34-11*	59-29*	67-14*	112-23*	
L\$MREV	5-77#						
L\$NAME	5-77#						
L\$PRIO	5-77#						
L\$PROT	5-77	100-8#					
L\$PRT	5-77#						
L\$REPP	5-77#						
L\$REV	5-77#						
L\$RPT	5-77	99-45#					
L\$SOFT	5-77	123-12	123-12#				
L\$SPC	5-77#						
L\$SPCP	5-77#						
L\$SPTP	5-77#						
L\$STA	5-77#						
L\$SW	5-77	8-10	8-10#				
L\$TEST	5-77#						
L\$TIML	5-77#						
L\$UNIT	5-77#	101-130	105-16	108-4			
L10000	7-10	7-27#					

L10001	8-10	8-27#					
L10002	27-17#						
L10003	27-22#						
L10004	27-28#						
L10005	27-32#						
L10006	27-37#						
L10007	27-49#						
L10010	27-54#						
L10011	27-59#						
L10012	27-63#						
L10013	27-67#						
L10014	27-71#						
L10015	27-75#						
L10016	27-79#						
L10017	27-83#						
L10020	67-48#						
L10021	81-14#						
L10022	82-22#						
L10023	95-9#						
L10024	96-3#						
L10025	99-118	99-156#					
L10027	108-45	109-15#					
L10030	110-18#						
L10031	111-30#						
L10032	113-6#						
L10033	112-40#						
L10034	112-43#						
L10035	112-46#						
L10036	112-50#						
L10037	112-56#						
L10040	112-66#						
L10041	117-24#						
L10042	118-14#						
L10043	119-117	120-189#					
L10044	121-50	122-7#					
L10045	123-12	124-11#					
L10046	126-14#						
L10050	126-14	126-21#					
LBUFE	23-23#	51-50*	51-53				
LBUFN	23-22#	51-47*	51-51	51-52*	51-53	51-62*	119-17
LBUFS	23-21#	31-15*	51-43	51-46*	119-10	119-13*	
LDDM	33-22#	33-31					
LDNEXT	33-24	33-27	33-29#				
LOAD	69-23	70-15#					
LOADB	68-43#	69-31					
LOADDM	33-26	68-14#					
LOADE1	68-56	70-26	71-3#				
LOADER	68-37	68-53	69-24	70-23	71-4#		
LOADM1	25-8#	71-3					
LOADT1	68-41	69-4#					
LOE	9-70#						
LOG	11-32#						
LOGCHK	119-11	119-22#					
LOGDAT	26-77#	51-60					
LOGFUL	26-78#	51-63					
LOGM1	26-79#	119-12					
LOGM2	26-80#	119-19					

LOGM3	26-81#	119-24							
LOGOUT	119-14#	119-18							
LOOP	27-61#	114-27							
LOOP00	25-40#								
LOOP01	25-41#	115-19							
LOOP02	25-42#	114-17	114-27						
LOOP03	26-41#	27-62	27-66						
LOOPA	27-65#	114-17							
LOT	9-70#								
LT11	69-12	69-14#							
LT1L1	69-10#	69-29							
LT1L2	69-18#	69-20							
MD.CMP	14-4#								
MD.CWB	14-22#								
MD.ERR	14-6#								
MD.EXP	14-5#								
MD.FEU	14-16#								
MD.IMF	14-20#								
MD.NXU	14-18#								
MD.PRI	14-23#								
MD.RIP	14-19#								
MD.SCH	14-7#								
MD.SCL	14-8#								
MD.SEC	14-9#								
MD.SEQ	14-14#								
MD.SER	14-10#								
MD.SPD	14-15#								
MD.SSH	14-11#								
MD.SWP	14-21#								
MD.VOL	14-17#								
MD.WBN	14-12#								
MD.WBV	14-13#								
MEMFIL	39-31#	39-34							
MESSAG	36-59	53-16#							
MESSG	26-58#	49-33	53-21						
MFG	5-48#								
MLDREM	25-11#	109-6							
MLDRER	107-15	109-6#							
MM.ACF	20-25#	77-19							
MM.ED	20-23#								
MM.EN	20-29#	79-23	80-24						
MM.PLF	20-20#	77-19							
MM.W	20-22#								
MSCP	11-31#								
MXFERE	49-21	49-36#							
MXFERP	26-39#	49-34							
MXFERX	49-25	49-27	49-30	49-35#					
NCON	67-45	67-47#							
NCONF	55-15	55-20#							
NCONS	55-14#	55-17							
NEWTAB	103-17	104-3#							
NOCLOC	26-71#	102-18							
NXMAD	23-9#	77-16*	77-37	81-12*	86-23*	86-28	112-29*	112-35	
NXMI	77-17	81-10#	86-24	112-30					
NXTTAB	103-11	104-21	105-10	105-15#					
OSAPTS	5-16#	5-77							
OSAU	5-16#	5-77							

P.MLUN	16-17#	16-33#			
P.MOD	15-20#				
P.OPCD	15-19#	16-6#	35-9	72-29*	
P.OTRF	15-27#	16-13#			
P.OVRL	15-51#	68-47*	68-48*		
P.RBN	15-36#				
P.RBNS	16-28#				
P.RCTC	16-29#				
P.RCTS	16-27#				
P.RGID	15-46#	70-21*			
P.RGOF	15-47#	70-20*			
P.SHST	16-23#	16-39#			
P.SHUN	15-33#	16-22#	16-38#		
P.STS	16-8#	35-16	68-55	70-25	
P.TIME	15-43#				
P.TRCK	16-24#				
P.UADR	15-23#	68-45*	70-18*	74-18*	
P.UNCL	16-40#				
P.UNFL	15-30#	16-18#	16-34#		
P.UNIT	15-18#	16-5#			
P.UNSZ	16-41#				
P.UNTI	16-20#	16-36#			
P.USEF	15-42#				
P.VRSN	15-39#	16-45#			
P.VSER	16-42#				
PAR0	20-3#	77-28*			
PAR1	20-4#	77-29*			
PAR2	20-5#	77-30*			
PAR3	20-6#	77-31*			
PAR4	20-7#	77-32*			
PAR5	20-8#	77-33*			
PAR6	20-9#	77-34*	79-20*	80-21*	
PAR7	20-10#	77-35*			
PAT16C	22-60#	42-19	119-64	119-65	
PAT16W	22-61#	119-66			
PDR0	20-12#	77-20*			
PDR1	20-13#	77-21*			
PDR2	20-14#	77-22*			
PDR3	20-15#	77-23*			
PDR4	20-16#	77-24*			
PDR5	20-17#	77-25*			
PDR6	20-18#	77-26*			
PDR7	20-19#	77-27*			
PNT	9-70#				
PNTERR	51-40	67-11#	119-15		
PNTNUM	56-25	56-31	56-37	61-14#	
PNTNUS	58-12	61-16#			
PORTST	112-42	114-1#			
POWDL C	101-79#	101-82	101-84		
POWDL L	101-87#				
POWDL Y	101-75	101-86#	101-90		
PRI	9-70#				
PRI00	9-70#	108-42	116-17	116-18	116-63
PRI01	9-70#				
PRI02	9-70#				
PRI03	9-70#				
PRI04	9-70#				

PRI05	9-70#							
PRI06	9-70#							
PRI07	5-77	9-70#	68-32	77-17	86-24	112-30	116-44	
RDDAT	93-34	93-36	93-54#					
RDDL	40-50	91-11#						
RDERR	93-37	93-107#						
RDREC	32-14	91-12	93-18#					
RDST	93-20	93-24#	93-28	93-48	93-50	93-52	93-86	93-94
READDM	31-25	32-13#						
README	32-15	32-20#						
RESET	31-14	97-10#	101-71	112-27	117-10			
RESETX	97-13	97-15#						
RESPCT	34-8#	34-63						
RESPDM	34-6#	34-64	40-62	112-63	117-16	118-12	119-27	119-110
RG.FLG	11-4#	73-24						
RG.OwN	11-3#	73-24	73-25					
RNTIME	67-38	98-12#						
RNTIMX	98-14	98-24#						
RPTCT	99-77#	99-116						
RPTCTN	99-78	99-84	99-114#					
RPTDT	99-83#	99-113						
RPTDTN	99-86	99-111#						
RPTEX	99-61	99-71#						
RPTMD1	99-107	99-152#						
RPTMD2	99-108	99-153#						
RPTMH2	99-75	99-150#						
RPTMS2	99-70	99-148#						
RPTMSD	99-88	99-151#						
RPTMSG	99-59	99-147#						
RPTMSH	99-74	99-149#						
RPTXX	99-73	99-117#						
RSPDRP	34-23	34-41	34-68#	35-12	35-19	35-26	36-7	36-19
RSPDSP	36-18	36-46#	36-62					
RSPIN	34-13	35-5#						
RSPMWR	35-7	35-7#						
RSPNRP	34-48	34-1	34-53	34-61#				
RSPNTO	34-34	34-37	34-39	34-42#				
RSPNXT	34-10	34-47#	34-70	36-42				
RSPOU	34-15	35-31#						
RSPOU2	36-26	36-35#						
RSPOU3	36-33	36-36#						
RSPOUT	35-31	36-24#						
RSPPK	25-4#	35-11						
RSPNE	25-6#	36-6						
RSPPRE	25-5#	35-25						
RSPPT2	36-3#							
RSPPT3	36-5	36-9#						
RSPPTW	35-24	35-30#						
RSPREF	35-17	35-23#						
RSPRPT	34-50	34-54#						
RSPSTE	25-7#	35-18						
RSPSTS	35-10	35-16#						
RSPTM	34-21	34-27#						
RSPTME	27-81#	34-22	75-44					
RSPTMM	26-73#	27-82						
RSPTMO	34-36	34-40#						
RSPTOE	27-77#	34-40	75-36					

	120-127	120-127	120-127	120-127	120-127	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131
	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131	120-131
	120-146	120-146	120-146	120-146	120-146	120-146	120-146	120-146	120-146	120-146	120-146	120-146	120-146	120-146
	120-150	120-150	120-150	120-150	120-150	120-150	120-150	120-150	120-150	120-150	120-150	120-150	120-150	120-150
	120-156	120-156	120-156	120-156	120-156	120-156	120-156	120-156	120-156	120-156	120-156	120-156	120-156	120-156
	120-160	120-160	120-160	120-160	120-160	120-160	120-160	120-160	120-160	120-160	120-160	120-160	120-160	120-160
	120-166	120-166	120-166	120-166	120-166	120-166	120-166	120-166	120-166	120-166	120-166	120-166	120-166	120-166
	120-180	120-180	120-180	120-180	120-180	120-180	120-180	120-180	120-180	120-180	120-180	120-180	120-180	120-180
	120-184	120-184	120-184	120-184	120-184	120-184	120-184	120-184	120-184	120-184	120-184	120-184	120-184	120-184
	121-50	122-1	122-1	122-1	122-1	122-1	122-1	122-1	122-1	122-1	122-1	122-1	122-1	122-1
	122-2	122-2	122-2	122-2	122-2	122-2	122-2	122-2	122-2	122-2	122-2	122-2	122-2	122-2
	122-3	122-3	122-3	122-3	122-3	122-3	122-3	122-3	122-3	122-3	122-3	122-3	122-3	122-3
	122-4	122-4	122-4	122-4	122-4	122-4	122-4	122-4	122-4	122-4	122-4	122-4	122-4	122-4
	122-5	122-5	122-5	122-5	122-5	122-5	122-5	122-5	122-5	122-5	122-5	122-5	122-5	122-5
	122-6	122-6	122-6	122-6	122-6	122-6	122-6	122-6	122-6	122-6	122-6	122-6	122-6	122-6
	123-12	124-1	124-1	124-1	124-1	124-1	124-1	124-1	124-1	124-1	124-1	124-1	124-1	124-1
	124-2	124-2	124-3	124-3	124-3	124-3	124-3	124-3	124-3	124-3	124-3	124-3	124-3	124-3
	124-3	124-3	124-3	124-4	124-4	124-4	124-4	124-4	124-4	124-4	124-4	124-4	124-4	124-4
	124-4	124-4	124-4	124-4	124-5	124-5	124-5	124-5	124-5	124-5	124-5	124-5	124-5	124-5
	124-6	124-6	124-6	124-6	124-6	124-6	124-6	124-6	124-6	124-6	124-6	124-6	124-6	124-6
	124-7	124-7	124-7	124-11	124-11	124-11	125-14	125-14	125-14	125-14	125-14	125-14	125-14	125-14
	125-14	126-14	126-14	126-14	126-14	126-14	126-14	126-14	126-14	126-14	126-14	126-14	126-14	126-14
SVCSUB	5-16#	5-24#	27-10#	27-87#	112-28	112-41	112-44	112-47	112-51	112-57				
SVCTAG	5-16#	5-26#	7-27	8-27	27-12#	27-17	27-17	27-17	27-22	27-22	27-22	27-28	27-28	27-28
	27-32	27-32	27-32	27-37	27-37	27-37	27-49	27-49	27-49	27-54	27-54	27-54	27-59	27-59
	27-59	27-63	27-63	27-63	27-67	27-67	27-67	27-71	27-71	27-71	27-75	27-75	27-75	27-79
	27-79	27-79	27-83	27-83	27-83	27-89#	41-66	67-48	81-14	82-22	95-9	96-3	99-156	108-32
	109-15	110-18	111-30	112-40	112-43	112-46	112-50	112-56	112-66	113-6	117-24	118-14	119-64	119-68
	120-14	120-23	120-29	120-32	120-33	120-36	120-41	120-50	120-51	120-54	120-57	120-59	120-83	120-121
SVCTST	120-127	120-131	120-146	120-150	120-156	120-160	120-166	120-180	120-184	120-189	122-7	124-11	126-14	126-21
TSSAUT	5-16#	5-23#	27-9#	27-86#	112-18	117-3	118-3	119-3						
TSSCLE	110-10#	110-18												
TSSDAT	111-8#	111-30												
TSSMAR	126-14	126-14#	126-21											
TSSHW	121-50	121-50#	122-7											
TSSINI	7-10	7-10#	7-27											
TSSMSG	101-8#	108-45	109-15											
	27-14#	27-17	27-19#	27-22	27-24#	27-28	27-30#	27-32	27-34#	27-37	27-39#	27-49	27-51#	27-54
	27-56#	27-59	27-61#	27-63	27-65#	27-67	27-69#	27-71	27-73#	27-75	27-77#	27-79	27-81#	27-83
	67-31#	67-48												
TSSPC	126-13#	126-22												
TSSPRO	100-8#													
TSSPTA	126-13#	126-14	126-14#											
TSSRPT	99-45#	99-118	99-156											
TSSSOF	123-12	123-12#	124-11											
TSSSRV	81-10#	81-14	82-18#	82-22	95-5#	95-9	96-1#	96-3						
TSSSUB	112-28#	112-40	112-41#	112-43	112-44#	112-46	112-47#	112-50	112-51#	112-56	112-57#	112-66		
TSSSW	8-10	8-10#	8-27											
TSSTES	112-18#	113-6	117-3#	117-24	118-3#	118-14	119-3#	119-117	120-189					
TSSARGC	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	27-15	27-15	27-15#	27-16	27-16	27-16	27-16#	27-16#	27-20	27-20	27-20#
	27-21	27-21	27-21	27-21#	27-21#	27-25	27-25	27-25#	27-26	27-26	27-26	27-26#	27-26#	27-27

	27-27	27-27	27-27#	27-27#	27-31	27-31	27-31	27-31#	27-31#	27-35	27-35	27-35#	27-36	27-36
	27-36	27-36#	27-36#	27-40	27-40	27-40#	27-41	27-41	27-41#	27-42	27-42	27-42#	27-45	27-45
	27-45	27-45	27-45#	27-45#	27-45#	27-52	27-52	27-52	27-52#	27-52#	27-53	27-53	27-53#	27-57
	27-57	27-57	27-57#	27-57#	27-58	27-58	27-58#	27-62	27-62	27-62	27-62	27-62#	27-62#	27-62#
	27-66	27-66	27-66	27-66	27-66#	27-66#	27-70	27-70	27-70	27-70	27-70#	27-70#	27-74	27-74
	27-74	27-74#	27-74#	27-78	27-78	27-78	27-78#	27-82	27-82	27-82	27-82	27-82	27-82#	27-82#
	27-82#	41-34	41-34	41-34#	41-42	41-42	41-42	41-42	41-42	41-42#	41-42#	41-42#	41-42#	41-43
	41-43	41-43#	41-44	41-44	41-44#	41-45	41-45	41-45	41-45#	41-53	41-53	41-53#	41-53#	41-59
	41-59	41-59	41-59#	41-59#	41-62	41-62	41-62#	41-113	41-113	41-113#	49-33	49-33	49-33	49-33
	49-33	49-33#	49-33#	49-33#	49-34	49-34	49-34#	51-60	51-60	51-60#	51-63	51-63	51-63	51-63#
	52-44	52-44	52-44	52-44#	53-21	53-21	53-21	53-21	53-21	53-21#	53-21#	53-21#	53-21#	53-21#
	55-18	55-18	55-18#	56-6	56-6	56-6	56-6#	56-14	56-14	56-14	56-14#	56-14#	56-14#	56-43
	56-43	56-43#	56-56	56-56	56-56#	56-64	56-64	56-64	56-64#	56-64#	58-13	58-13	58-13#	58-20
	58-20	58-20#	61-59	61-59	61-59	61-59#	61-69	61-69	61-69	61-69	61-69#	61-69#	64-39	64-39
	64-39#	64-43	64-43	64-43#	65-4	65-4	65-4#	65-4#	65-4#	65-13	65-13	65-13	65-13#	65-13#
	65-30	65-30	65-30	65-30#	65-30#	67-34	67-34	67-34	67-34	67-34	67-34#	67-34#	67-34#	67-34#
	67-37	67-37	67-37	67-37#	67-37#	98-23	98-23	98-23	98-23	98-23	98-23#	98-23#	98-23#	98-23#
	98-24	98-24	98-24#	99-59	99-59	99-59	99-59#	99-70	99-70	99-70	99-70	99-70	99-70	99-70#
	99-70#	99-70#	99-70#	99-71	99-71	99-71#	99-74	99-74	99-74#	99-75	99-75	99-75#	99-88	99-88
	99-88	99-88	99-88#	99-88#	99-88#	99-107	99-107	99-107	99-107	99-107	99-107	99-107#	99-107#	99-107#
	99-107#	99-107#	99-108	99-108	99-108	99-108	99-108	99-108#	99-108#	99-108#	99-108#	102-18	102-18	102-18#
	108-12	108-12	108-12#	108-21	108-21	108-21	108-21	108-21	108-21#	108-21#	108-21#	108-21#	116-16	116-16
	116-16	116-16	116-16#	116-16#	116-16#	116-69	116-69	116-69#	119-12	119-12	119-12#	119-19	119-19	119-19#
	119-24	119-24	119-24#	119-76	119-76	119-76#	120-12	120-12	120-12	120-12	120-12	120-12#	120-12#	120-12#
	120-12#	120-77	120-77	120-77#	120-78	120-78	120-78#	120-79	120-79	120-79#	120-80	120-80	120-80#	120-81
T\$CODE	120-81	120-81#	120-82	120-82	120-82#									
	41-66#	41-66	41-66	41-66#	41-66#	41-66#	108-32	108-32	108-32	108-32#	108-32#	108-32#	119-64	119-64
	119-64	119-64#	119-64#	119-64#	119-68	119-68	119-68	119-68#	119-68#	119-68#	120-14	120-14	120-14	120-14#
	120-14#	120-14#	120-23	120-23	120-23	120-23#	120-23#	120-23#	120-29	120-29	120-29	120-29#	120-29#	120-29#
	120-32	120-32	120-32	120-32#	120-32#	120-32#	120-33	120-33	120-33	120-33#	120-33#	120-33#	120-36	120-36
	120-36	120-36#	120-36#	120-36#	120-41	120-41	120-41	120-41#	120-41#	120-41#	120-50	120-50	120-50	120-50#
	120-50#	120-50#	120-51	120-51	120-51	120-51#	120-51#	120-51#	120-54	120-54	120-54	120-54#	120-54#	120-54#
	120-57	120-57	120-57	120-57#	120-57#	120-57#	120-59	120-59	120-59	120-59	120-59#	120-59#	120-83	120-83
	120-83	120-83#	120-83#	120-83#	120-121	120-121	120-121	120-121#	120-121#	120-121#	120-127	120-127	120-127	120-127#
	120-127#	120-127#	120-131	120-131	120-131	120-131#	120-131#	120-131#	120-146	120-146	120-146	120-146#	120-146#	120-146#
	120-150	120-150	120-150	120-150#	120-150#	120-150#	120-156	120-156	120-156	120-156#	120-156#	120-156#	120-160	120-160
	120-160	120-160#	120-160#	120-160#	120-166	120-166	120-166	120-166#	120-166#	120-166#	120-180	120-180	120-180	120-180#
	120-180#	120-180#	120-184	120-184	120-184	120-184#	120-184#	120-184#	122-1	122-1	122-1	122-1#	122-1#	122-1#
	122-2	122-2	122-2	122-2#	122-2#	122-2#	122-3	122-3	122-3	122-3#	122-3#	122-3#	122-4	122-4
	122-4	122-4#	122-4#	122-4#	122-5	122-5	122-5	122-5#	122-5#	122-5#	122-6	122-6	122-6	122-6#
	122-6#	122-6#	124-1	124-1	124-1	124-1#	124-1#	124-1#	124-2	124-2#	124-3	124-3	124-3	124-3#
	124-3#	124-3#	124-4	124-4	124-4	124-4#	124-4#	124-4#	124-5	124-5	124-5	124-5#	124-5#	124-5#
	124-6	124-6	124-6	124-6#	124-6#	124-6#	124-7	124-7	124-7	124-7#	124-7#	124-7#		
T\$ERRN	5-16#	28-7	28-7#	32-20	32-20#	34-22	34-22#	34-40	34-40#	35-11	35-11#	35-18	35-18#	35-25
	35-25#	36-6	36-6#	59-24	59-24#	71-3	71-3#	75-36	75-36#	75-44	75-44#	85-18	85-18#	85-37
	85-37#	86-30	86-30#	87-44	87-44#	88-36	88-36#	89-5	89-5#	94-1	94-1#	94-3	94-3#	101-122
	101-122#	109-2	109-2#	109-6	109-6#	109-11	109-11#	112-37	112-37#	114-17	114-17#	114-27	114-27#	115-19
	115-19#	116-41	116-41#	116-61	116-61#	116-67	116-67#							
T\$EXCP	41-66	41-66#	119-64	119-64#	119-68	119-68#	120-14	120-14#	120-23	120-23#	120-41	120-41#	120-83	120-83#
	120-121	120-121#	120-127	120-127#	120-131	120-131#	120-146	120-146#	120-150	120-150#	120-156	120-156#	120-160	120-160#
	120-180	120-180#	120-184	120-184#	122-1	122-1#	122-2	122-2#	122-3	122-3#	122-4	122-4#	122-5	122-5#
	124-3	124-3#	124-4	124-4#										
T\$FLAG	99-118	99-118#	99-118#	108-45	108-45	108-45#	108-45#	119-117	119-117	119-117#	119-117#			
T\$FREE	125-14	126-22#												
T\$GMAN	5-16#	41-66	41-66#	41-66#	119-64#	119-64#	119-68#	119-68#	120-14#	120-14#	120-23	120-23#	120-23#	120-41#
	120-41#	120-83#	120-83#	120-121#	120-121#	120-127	120-127#	120-127#	120-131	120-131#	120-131#	120-146#	120-146#	120-150#
	120-150#	120-156#	120-156#	120-160#	120-160#	120-180	120-180#	120-180#	120-184	120-184#	120-184#			

T\$HILI	41-66	41-66#	119-64	119-64#	119-68	119-68#	120-14	120-14#	120-23	120-23#	120-41	120-41#	120-83	120-83#
	120-121	120-121#	120-127	120-127#	120-131	120-131#	120-146	120-146#	120-150	120-150#	120-156	120-156#	120-160	120-160#
	120-180	120-180#	120-184	120-184#	122-1	122-1#	122-2	122-2#	122-3	122-3#	122-4	122-4#	122-5	122-5#
	124-3	124-3#	124-4	124-4#										
T\$LAST	5-16#	125-14#	126-13											
T\$LOLI	41-66	41-66#	119-64	119-64#	119-68	119-68#	120-14	120-14#	120-23	120-23#	120-41	120-41#	120-83	120-83#
	120-121	120-121#	120-127	120-127#	120-131	120-131#	120-146	120-146#	120-150	120-150#	120-156	120-156#	120-160	120-160#
	120-180	120-180#	120-184	120-184#	122-1	122-1#	122-2	122-2#	122-3	122-3#	122-4	122-4#	122-5	122-5#
	124-3	124-3#	124-4	124-4#										
T\$LSYM	5-16	5-16#	7-27	8-27	27-17	27-22	27-28	27-32	27-37	27-49	27-54	27-59	27-63	27-67
	27-71	27-75	27-79	27-83	67-48	81-14	82-22	95-9	96-3	99-156	109-15	110-18	111-30	112-40
	112-43	112-46	112-50	112-56	112-66	113-6	117-24	118-14	120-189	122-7	124-11			
T\$LTNO	125-14#													
T\$NEST	5-16#	5-43	5-43	5-43#	7-10	7-10	7-10#	7-27	7-27	7-27	7-27#	8-10	8-10	8-10#
	8-27	8-27	8-27	8-27#	8-29	8-29	8-29#	8-29#	9-48	9-48	9-48#	27-14	27-14	27-14#
	27-17	27-17	27-17	27-17#	27-19	27-19	27-19#	27-22	27-22	27-22	27-22#	27-24	27-24	27-24#
	27-28	27-28	27-28	27-28#	27-30	27-30	27-30#	27-32	27-32	27-32	27-32#	27-34	27-34	27-34#
	27-37	27-37	27-37	27-37#	27-39	27-39	27-39#	27-49	27-49	27-49	27-49#	27-51	27-51	27-51#
	27-54	27-54	27-54	27-54#	27-56	27-56	27-56#	27-59	27-59	27-59	27-59#	27-61	27-61	27-61#
	27-63	27-63	27-63	27-63#	27-65	27-65	27-65#	27-67	27-67	27-67	27-67#	27-69	27-69	27-69#
	27-71	27-71	27-71	27-71#	27-73	27-73	27-73#	27-75	27-75	27-75	27-75#	27-77	27-77	27-77#
	27-79	27-79	27-79	27-79#	27-81	27-81	27-81#	27-83	27-83	27-83	27-83#	67-31	67-31	67-31#
	67-48	67-48	67-48	67-48#	81-10	81-10	81-10#	81-14	81-14	81-14	81-14#	82-18	82-18	82-18#
	82-22	82-22	82-22	82-22#	95-5	95-5	95-5#	95-9	95-9	95-9	95-9#	96-1	96-1	96-1#
	96-3	96-3	96-3	96-3#	98-28	98-28	98-28#	98-28#	99-38	99-38	99-38#	99-45	99-45	99-45#
	99-156	99-156	99-156	99-156#	100-8	100-8	100-8#	100-14	100-14	100-14	100-14#	101-8	101-8	101-8#
	109-15	109-15	109-15	109-15#	110-10	110-10	110-10#	110-18	110-18	110-18	110-18#	111-8	111-8	111-8#
	111-30	111-30	111-30	111-30#	111-31	111-31	111-31#	111-31#	112-16	112-16	112-16#	112-18	112-18	112-18#
	112-28	112-28	112-28#	112-40	112-40	112-40#	112-40#	112-41	112-41	112-41#	112-43	112-43	112-43	112-43#
	112-44	112-44	112-44#	112-46	112-46	112-46#	112-46#	112-47	112-47	112-47#	112-50	112-50	112-50	112-50#
	112-51	112-51	112-51#	112-56	112-56	112-56#	112-56#	112-57	112-57	112-57#	112-66	112-66	112-66	112-66#
	113-6	113-6	113-6	113-6#	117-3	117-3	117-3#	117-24	117-24	117-24#	117-24#	118-3	118-3	118-3#
	118-14	118-14	118-14	118-14#	119-3	119-3	119-3#	120-189	120-189	120-189	120-189#	120-190	120-190	120-190
	120-190#	121-39	121-39	121-39#	121-50	121-50	121-50#	122-7	122-7	122-7	122-7#	123-12	123-12	123-12#
	124-11	124-11	124-11	124-11#	125-15	125-15	125-15#	125-15#	125-15#	125-15#	125-15#	125-15#	125-15#	125-15#
T\$NSO	5-43#	8-29	9-48#	98-28	99-38#	111-31	112-16#	120-190	121-39#	125-15				
T\$NS1	7-10#	7-27	8-10#	8-27	27-14#	27-17	27-19#	27-22	27-24#	27-28	27-30#	27-32	27-34#	27-37
	27-39#	27-49	27-51#	27-54	27-56#	27-59	27-61#	27-63	27-65#	27-67	27-69#	27-71	27-73#	27-75
	27-77#	27-79	27-81#	27-83	67-31#	67-48	81-10#	81-14	82-18#	82-22	95-5#	95-9	96-1#	96-3
	99-45#	99-156	100-8#	100-14	101-8#	109-15	110-10#	110-18	111-8#	111-30	112-18#	113-6	117-3#	117-24
	118-3#	118-14	119-3#	120-189	121-50#	122-7	123-12#	124-11						
T\$NS2	112-28#	112-40	112-41#	112-43	112-44#	112-46	112-47#	112-50	112-51#	112-56	112-57#	112-66		
T\$PCNT	126-13#	126-14	126-14	126-14#										
T\$PTAB	126-14	126-14#												
T\$PTHV	5-77	126-22#												
T\$PTNU	5-16#	126-14	126-14#	126-22	126-22									
T\$SAVL	5-16#													
T\$SEGL	5-16#													
T\$SIZE	125-14	126-22#												
T\$SUBN	5-16#	112-18#	112-28	112-28	112-28#	112-41	112-41	112-41#	112-44	112-44	112-44#	112-47	112-47	112-47#
	112-51	112-51	112-51#	112-57	112-57	112-57#	117-3#	118-3#	119-3#					
T\$TAGL	5-16#													
T\$TAGN	5-16#	7-10	7-10	7-10#	8-10	8-10	8-10#	27-14	27-14	27-14#	27-19	27-19	27-19#	27-24
	27-24	27-24#	27-30	27-30	27-30#	27-34	27-34	27-34#	27-39	27-39	27-39#	27-51	27-51	27-51#
	27-56	27-56	27-56#	27-61	27-61	27-61#	27-65	27-65	27-65#	27-69	27-69	27-69#	27-73	27-73
	27-73#	27-77	27-77	27-77#	27-81	27-81	27-81#	67-31	67-31	67-31#	81-10	81-10	81-10#	82-18
	82-18	82-18#	95-5	95-5	95-5#	96-1	96-1	96-1#	99-45	99-45	99-45#	100-8	100-8	100-8#

T1ERR	113-5#								
T1GOOD	112-36	112-39#							
T1MSIZ	36-46	39-26#							
T1NAME	24-32#	56-75							
T1NEXT	112-22#	113-3							
T1SKIP	112-24	112-38	113-1#						
T2	6-8	117-3#							
T2CMD	36-48	41-29#							
T2CMD0	41-33	41-36#							
T2CMD2	41-38	41-50#							
T2CMD3	41-72	41-84#							
T2CMD9	41-34#								
T2CMDE	41-79	41-88	41-91	41-93	41-108	41-113#	66-47		
T2CMDM	41-30	41-32#							
T2CMDN	41-98	41-100#							
T2CMDQ	41-51	41-66#	41-114						
T2CMDR	41-86	41-95#							
T2CMDV	41-70	41-78#							
T2CMDW	41-101	41-105	41-107#						
T2CMDX	41-31	41-35	41-80	41-111#					
T2CMS1	26-61#	41-42							
T2CMS2	26-62#	41-43							
T2CMS3	26-65#	41-44							
T2CMS4	26-67#	41-45							
T2CMS5	26-70#	41-113							
T2DLL	36-47	40-37#							
T2DR	23-16#	41-48*	41-76*	41-77					
T2GND1	66-16	66-19#							
T2GND2	66-20#	66-41							
T2GND3	66-25	66-31#							
T2GNE	66-23	66-27	66-29	66-45#					
T2GNUM	41-74	41-78	41-90	41-100	41-104	41-107	66-12#	66-18	
T2GNX	66-14	66-39	66-42#						
T2NAME	24-33#	56-76							
T2NEXT	117-10#	117-22							
T2PNT	65-7	65-9	65-17#						
T2PNTB	41-61	65-12#							
T2PNTD	65-27	65-29#							
T2PNT0	65-22	65-24#							
T2PNTW	41-55	41-57	65-3#						
T2WARN	26-60#	41-34							
T2WRO	23-15#	41-47*	41-56	41-96	41-110*				
T2WRR	23-14#	41-46*	41-54	41-95	41-109*				
T3	6-8	118-3#							
T3NAME	24-34#	56-77							
T4	6-8	119-3#							
T4BB	25-14#	120-14							
T4BB1	36-51	45-22#							
T4BB1E	45-24	45-30#							
T4BB2	36-52	46-15#							
T4BB2E	46-17	46-23#							
T4BBI	25-15#	120-23							
T4BE	25-27#	120-121							
T4BEG	25-28#	120-127							
T4CON	119-20	119-23	119-25#						
T4CYL	25-34#	120-166							
T4CYLB	25-35#	120-180							

T4CYLE	25-36#	120-184			
T4DCA	25-22#	120-51			
T4DCR	25-23#	120-54			
T4DEF	119-35	119-80#			
T4DEFA	119-81#	119-97			
T4DEFB	119-84#	119-94			
T4DEFC	119-90#	119-92			
T4DEFD	119-93#				
T4DEFE	119-85	119-95#			
T4DEFW	119-37	119-76#			
T4DP	25-20#	120-41			
T4DPC	25-37#	119-64			
T4DPD	25-38#	119-68			
T4ECC	25-21#	120-50			
T4END	25-29#	120-131			
T4EXIT	119-112	119-116#			
T4GRC	25-32#	120-156			
T4GRP	25-33#	120-160			
T4MPRM	36-49	42-18#			
T4MXFR	36-55	49-18#			
T4NAME	24-35#	56-78			
T4OPT1	26-50#	120-77			
T4OPT2	26-51#	120-78			
T4OPT3	26-52#	120-79			
T4OPT4	26-53#	120-80			
T4OPT5	26-54#	120-81			
T4OPT6	26-55#	120-82			
T4OPT7	25-26#	41-66	120-83		
T4PRM1	119-43#	119-58			
T4PRM2	119-46#	119-55			
T4PRM3	119-49	119-52	119-54#		
T4PRM4	119-47	119-56#			
T4PRM5	119-67#	119-71			
T4Q01	120-22#	120-25	120-27		
T4Q02	120-16	120-28#			
T4Q03	120-32#				
T4Q04	120-35	120-38#			
T4Q05	120-37	120-39#			
T4Q06	120-31	120-44#			
T4Q07	120-45	120-47	120-50#		
T4Q08	120-53	120-56#			
T4Q09	120-49	120-55	120-57#		
T4Q10	120-65	120-68#			
T4Q11	120-67	120-69	120-72	120-75	120-77#
T4Q12	120-85	120-88#			
T4Q13	120-89	120-95#			
T4Q14	120-98	120-102#			
T4Q15	120-105	120-107#			
T4Q16	120-94	120-108	120-111#		
T4Q17	120-112#				
T4Q18	120-115#	120-117			
T4Q19	120-87	120-91	120-101	120-110	120-118#
T4Q20	120-126#	120-129	120-135		
T4Q21	120-130#	120-133			
T4Q22	120-119	120-137#			
T4Q23	120-149#	120-153			
T4Q24	120-144	120-155#			

.BR	21-31#													
AND	21-3#	68-29	107-25	119-87	119-102									
ASSUME	21-39#	33-25	50-23	99-79	99-87	99-109	99-110	103-19	103-20	112-25	114-6	116-6		
BCOMPL	41-33	101-92	102-14	102-16										
BGNAUT	110-10													
BGNCLN	111-8													
BGNHRD	121-50													
BGNHW	7-10													
BGNINI	101-8													
BGNMOD	5-43	9-48	99-38	112-16	121-39									
BGNMSG	27-14	27-19	27-24	27-30	27-34	27-39	27-51	27-56	27-61	27-65	27-69	27-73	27-77	27-81
	67-31													
BGNPRO	100-8													
BGNPTA	126-14													
BGNRPT	99-45													
BGNSET	126-13													
BGNSFT	123-12													
BGNSRV	81-10	82-18	95-5	96-1										
BGNSUB	112-28	112-41	112-44	112-47	112-51	112-57								
BGNSW	8-10													
BGNTST	112-18	117-3	118-3	119-3										
BNCOMP	101-64	101-67	101-70	101-115	103-11	106-6	108-31	119-37						
BREAK	34-43	75-28	86-13	88-27	93-24	97-10	101-79	101-88	115-11	116-33	119-113			
BRESET	97-11	102-1												
BUILD	56-84#	56-103	56-115											
CLOCK	102-13	102-15												
CLOSE	92-12													
CLRVEC	77-36	86-27	112-34	116-72	116-77									
DELAY	101-87													
DESCRI	24-22													
DEVTYP	24-12													
DISPAT	6-8													
DISPLA	124-2													
DOCLN	28-8	32-21	94-2	94-4	101-123	108-35	109-3	109-7	109-12					
DORPT	34-54	101-73	119-116											
ENDAUT	110-18													
ENDCLN	111-30													
ENDHRD	122-7													
ENDHW	7-27													
ENDINI	109-15													
ENDMOD	8-29	98-28	111-31	120-190	125-15									
ENDMSG	27-17	27-22	27-28	27-32	27-37	27-49	27-54	27-59	27-63	27-67	27-71	27-75	27-79	27-83
	67-48													
ENDPRO	100-14													
ENDPTA	126-21													
ENDRPT	99-156													
ENDSET	126-22													
ENDSFT	124-11													
ENDSRV	81-14	82-22	95-9	96-3										
ENDSUB	112-40	112-43	112-46	112-50	112-56	112-66								
ENDSW	8-27													
ENDTST	113-6	117-24	118-14	120-189										
ENTRY	56-97#	56-103	56-103	56-103	56-103	56-103	56-103	56-103	56-103	56-103	56-109#	56-115	56-115	56-115
	56-115	56-115	56-115	56-115										
EQUALS	9-70													
ERRDF	34-22	34-40	59-24	75-36	75-44	85-18	85-37	86-30	87-44	88-36	89-5	112-37	114-17	114-27

ERRHRD	115-19 35-11	116-41 35-18	116-61 35-25	116-67 36-6	71-3																							
ERROR	67-27																											
ERRSF	28-7	32-20	94-1	94-3	101-122	109-2	109-6	109-11																				
ERRTBL	22-21																											
EXIT	99-118	108-45	119-117																									
GETBYT	93-26	93-31	93-41	93-76	93-83	93-89	93-99	93-102																				
GMANID	41-66 120-180	119-64 120-184	119-68	120-14	120-23	120-41	120-83	120-121	120-127	120-131	120-146	120-150	120-156	120-160														
GMANIL	108-32	120-29	120-32	120-33	120-36	120-50	120-51	120-54	120-57	120-59	120-166																	
GPHARD	101-114	103-10	106-5																									
GPRMA	122-1	122-2																										
GPRMD	41-66 120-121 120-180	41-66# 120-121# 120-180#	119-64 120-127 120-184	119-64# 120-127# 120-184#	119-68 120-131 122-3	119-68# 120-131# 122-4	120-14 120-146 122-5	120-14# 120-146# 124-3	120-23 120-150 124-4	120-23# 120-150# 124-4	120-41 120-156 124-5	120-41# 120-156# 124-6	120-83 120-160 124-7	120-83# 120-160#														
GPRML	108-32 120-54	108-32# 120-54#	120-29 120-57	120-29# 120-57#	120-32 120-59	120-32# 120-59#	120-33 120-166	120-33# 120-166#	120-36 122-6	120-36# 124-1	120-50 124-5	120-50# 124-6	120-51 124-7	120-51#														
HEADER	5-77																											
LASTAD	125-14																											
MSBYTE	5-77	5-77	5-77	5-77#																								
MSCHEC	99-118	99-118#	108-45	108-45#	119-117	119-117#																						
MSCNTO	41-66 120-32 120-57 120-150 122-2 124-4	41-66# 120-32# 120-57# 120-150# 122-2# 124-4#	108-32 120-33 120-59 120-156 122-3 124-5	108-32# 120-33# 120-59# 120-156# 122-3# 124-5#	119-64 120-36 120-83 120-160 122-4 124-6	119-64# 120-36# 120-83# 120-160# 122-4# 124-6#	119-68 120-41 120-121 120-166 122-5 124-7	119-68# 120-41# 120-121# 120-166# 122-5# 124-7#	120-14 120-50 120-127 120-180 122-6	120-14# 120-50# 120-127# 120-180# 122-6#	120-23 120-51 120-131 120-184 124-1	120-23# 120-51# 120-131# 120-184# 124-1#	120-29 120-54 120-146 122-1 124-3	120-29# 120-54# 120-146# 122-1# 124-3#														
MSCOUN	27-15 27-31 27-45# 27-66# 41-42 41-113 53-21 56-64 65-4 98-23 99-75 102-18 119-12# 120-78#	27-15# 27-31# 27-52 27-70 41-42# 41-113# 53-21 56-64# 65-4# 98-23# 99-75# 102-18# 119-19 120-79	27-16 27-35 27-52# 27-70# 41-43 49-33 53-21 58-13 65-13 98-24 99-88 108-12 119-19# 120-79#	27-16# 27-35# 27-53 27-74 41-43# 49-33 53-21# 58-13# 65-13# 98-24# 99-88 108-12# 119-24 120-80	27-20 27-36 27-53# 27-74# 41-44 49-33 55-18 58-20 65-30 99-59 99-88# 108-21 119-24# 120-80#	27-20# 27-36# 27-57 27-78 41-44# 49-33# 55-18# 58-20# 65-30# 99-59# 99-88# 108-21 119-76 120-81	27-21 27-40 27-57# 27-78# 41-45 49-34 56-6 61-59 67-34 99-70 99-107 108-21 119-76# 120-81#	27-21# 27-40# 27-58 27-78 41-45# 49-34# 56-6# 61-59# 67-34 99-70 99-107 108-21# 119-76# 120-81#	27-25 27-41 27-58# 27-82 41-53 51-60 56-14 61-69 67-34 99-70 99-107 116-16 120-12 120-82	27-25# 27-41# 27-58# 27-82# 41-53# 51-60# 56-14# 61-69# 67-34# 99-70# 99-107# 116-16 120-12 120-82#	27-26 27-42 27-62 27-82# 41-59 51-63 56-43 64-39 67-37 99-71 99-108 116-16# 120-12# 120-82#	27-26# 27-42# 27-62# 27-82# 41-59# 51-63# 56-43# 64-39# 67-37# 99-71# 99-108 116-16# 120-12# 120-82#	27-27 27-45 27-66 41-42 41-62 52-44 56-56 64-43 98-23 99-74 99-108 116-69 120-77 5-77 5-77 5-77	27-27# 27-45# 27-66# 41-42# 41-62# 52-44# 56-56# 64-43# 98-23# 99-74# 99-108# 116-69# 120-77# 5-77 5-77# 5-77#	5-77 5-77 5-77 24-22# 7-27 27-37 27-75 96-3 111-31 113-6 125-15 41-66 120-32 120-57 120-150 122-2	5-77 5-77 5-77 24-22# 7-27# 27-37# 27-75# 96-3# 111-31# 113-6# 125-15# 41-66# 120-32# 120-57# 120-150# 122-2#	8-27 27-49 27-79 98-28 112-40 117-24 126-14 108-32 120-33 120-59 120-156 122-3	8-27# 27-49# 27-79# 98-28# 112-40# 117-24# 126-14# 108-32# 120-33# 120-59# 120-156# 122-3#	8-29 27-54 27-83 99-156 112-43 118-14 119-64 120-36 120-83 120-160 122-4	8-29# 27-54# 27-83# 99-156# 112-43# 118-14# 119-64# 120-36# 120-83# 120-160# 122-4#	27-17 27-59 67-48 100-14 112-46 120-189 119-68 120-41 120-121 120-166 122-5	27-17# 27-59# 67-48# 100-14# 112-46# 120-189# 119-68# 120-41# 120-121# 120-166# 122-5#	27-22 27-63 81-14 109-15 112-50 120-190 120-14 120-50 120-127 120-180 122-6	27-22# 27-63# 81-14# 109-15# 112-50# 120-190# 120-14# 120-50# 120-127# 120-180# 122-6#	27-28 27-67 82-22 110-18 112-56 122-7 120-23 120-51 120-131 120-184 124-1	27-28# 27-67# 82-22# 110-18# 112-56# 122-7# 120-23# 120-51# 120-131# 120-184# 124-1#	27-32 27-71 95-9 111-30 112-66 124-11 120-29 120-54 120-146 122-1 124-3	27-32# 27-71# 95-9# 111-30# 112-66# 124-11# 120-29# 120-54# 120-146# 122-1# 124-3#
MSDATA	5-77 5-77 5-77	5-77 5-77 5-77	5-77 5-77 5-77	5-77 5-77 5-77	5-77 5-77 5-77	5-77 5-77 5-77	5-77 5-77 5-77	5-77 5-77 5-77	5-77 5-77 5-77#	5-77 5-77 5-77#	5-77 5-77 24-12	5-77 5-77 24-12#	5-77 5-77 24-22															
MSDECR	24-22# 7-27 27-37 27-75 96-3 111-31 113-6 125-15 41-66 120-32 120-57 120-150 122-2	7-27# 27-37# 27-75# 96-3# 111-31# 113-6# 125-15# 41-66# 120-32# 120-57# 120-150# 122-2#	8-27 27-49 27-79 98-28 112-40 117-24 126-14 108-32 120-33 120-59 120-156 122-3	8-27# 27-49# 27-79# 98-28# 112-40# 117-24# 126-14# 108-32# 120-33# 120-59# 120-156# 122-3#	8-29 27-54 27-83 99-156 112-43 118-14 119-64 120-36 120-83 120-160 122-4	8-29# 27-54# 27-83# 99-156# 112-43# 118-14# 119-64# 120-36# 120-83# 120-160# 122-4#	27-17 27-59 67-48 100-14 112-46 120-189 119-68 120-41 120-121 120-166 122-5	27-17# 27-59# 67-48# 100-14# 112-46# 120-189# 119-68# 120-41# 120-121# 120-166# 122-5#	27-22 27-63 81-14 109-15 112-50 120-190 120-14 120-50 120-127 120-180 122-6	27-22# 27-63# 81-14# 109-15# 112-50# 120-190# 120-14# 120-50# 120-127# 120-180# 122-6#	27-28 27-67 82-22 110-18 112-56 122-7 120-23 120-51 120-131 120-184 124-1	27-28# 27-67# 82-22# 110-18# 112-56# 122-7# 120-23# 120-51# 120-131# 120-184# 124-1#	27-32 27-71 95-9 111-30 112-66 124-11 120-29 120-54 120-146 122-1 124-3	27-32# 27-71# 95-9# 111-30# 112-66# 124-11# 120-29# 120-54# 120-146# 122-1# 124-3#														
MSDEFA	41-66 120-32 120-57 120-150 122-2	41-66# 120-32# 120-57# 120-150# 122-2#	108-32 120-33 120-59 120-156 122-3	108-32# 120-33# 120-59# 120-156# 122-3#	119-64 120-36 120-83 120-160 122-4	119-64# 120-36# 120-83# 120-160# 122-4#	119-68 120-41 120-121 120-166 122-5	119-68# 120-41# 120-121# 120-166# 122-5#	120-14 120-50 120-127 120-180 122-6	120-14# 120-50# 120-127# 120-180# 122-6#	120-23 120-51 120-131 120-184 124-1	120-23# 120-51# 120-131# 120-184# 124-1#	120-29 120-54 120-146 122-1 124-3	120-29# 120-54# 120-146# 122-1# 124-3#														

	124-4	124-4#	124-5	124-5#	124-6	124-6#	124-7	124-7#						
M\$ENDE	7-27#	8-27#	8-29#	27-17#	27-22#	27-28#	27-32#	27-37#	27-49#	27-54#	27-59#	27-63#	27-67#	27-71#
	27-75#	27-79#	27-83#	67-48#	81-14#	82-22#	95-9#	96-3#	98-28#	99-156#	109-15#	110-18#	111-30#	111-31#
M\$ERRI	112-40#	112-43#	112-46#	112-50#	112-56#	112-66#	113-6#	117-24#	118-14#	120-189#	120-190#	122-7#	124-11#	125-15#
	28-7	28-7#	32-20	32-20#	34-22	34-22#	34-40	34-40#	35-11	35-11#	35-18	35-18#	35-25	35-25#
	36-6	36-6#	59-24	59-24#	71-3	71-3#	75-36	75-36#	75-44	75-44#	85-18	85-18#	85-37	85-37#
	86-30	86-30#	87-44	87-44#	88-36	88-36#	89-5	89-5#	94-1	94-1#	94-3	94-3#	101-122	101-122#
	109-2	109-2#	109-6	109-6#	109-11	109-11#	112-37	112-37#	114-17	114-17#	114-27	114-27#	115-19	115-19#
	116-41	116-41#	116-61	116-61#	116-67	116-67#								
M\$EXCP	41-66	41-66	41-66#	119-64	119-64	119-64#	119-68	119-68	119-68#	120-14	120-14	120-14#	120-23	120-23
	120-23#	120-41	120-41	120-41#	120-83	120-83	120-83#	120-121	120-121	120-121#	120-127	120-127	120-127#	120-131
	120-131	120-131#	120-146	120-146	120-146#	120-150	120-150	120-150#	120-156	120-156	120-156#	120-160	120-160	120-160#
	120-180	120-180	120-180#	120-184	120-184	120-184#	122-1	122-1	122-1#	122-2	122-2	122-2#	122-3	122-3
	122-3#	122-4	122-4	122-4#	122-5	122-5	122-5#	124-3	124-3	124-3#	124-4	124-4	124-4#	
M\$EXIT	99-118#	108-45	108-45#	119-117	119-117#									
M\$EXSE	99-118#	108-45#	119-117#											
M\$EXTJ	99-118	99-118#	108-45#	119-117#										
M\$GEN	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77
	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77
	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	6-8	6-8#	7-10	7-10	7-10#	7-10#	7-27	7-27#
	8-10	8-10	8-10#	8-10#	8-27	8-27#	22-21	22-21#	24-12	24-12#	24-22	24-22#	27-14	27-14#
	27-17	27-17#	27-19	27-19#	27-22	27-22#	27-24	27-24#	27-28	27-28#	27-30	27-30#	27-32	27-32#
	27-34	27-34#	27-37	27-37#	27-39	27-39#	27-49	27-49#	27-51	27-51#	27-54	27-54#	27-56	27-56#
	27-59	27-59#	27-61	27-61#	27-63	27-63#	27-65	27-65#	27-67	27-67#	27-69	27-69#	27-71	27-71#
	27-73	27-73#	27-75	27-75#	27-77	27-77#	27-79	27-79#	27-81	27-81#	27-83	27-83#	41-66	41-66#
	67-31	67-31#	67-48	67-48#	81-10	81-10#	81-14	81-14#	82-18	82-18#	82-22	82-22#	95-5	95-5#
	95-9	95-9#	96-1	96-1#	96-3	96-3#	99-45	99-45#	99-156	99-156#	100-8	100-8#	101-8	101-8#
	108-32	108-32#	109-15	109-15#	110-10	110-10#	110-18	110-18#	111-8	111-8#	111-30	111-30#	112-18	112-18#
	112-28	112-28#	112-40	112-40#	112-41	112-41#	112-43	112-43#	112-44	112-44#	112-46	112-46#	112-47	112-47#
	112-50	112-50#	112-51	112-51#	112-56	112-56#	112-57	112-57#	112-66	112-66#	113-6	113-6#	117-3	117-3#
	117-24	117-24#	118-3	118-3#	118-14	118-14#	119-3	119-3#	119-64	119-64#	119-68	119-68#	120-14	120-14#
	120-23	120-23#	120-29	120-29#	120-32	120-32#	120-33	120-33#	120-36	120-36#	120-41	120-41#	120-50	120-50#
	120-51	120-51#	120-54	120-54#	120-57	120-57#	120-59	120-59#	120-83	120-83#	120-121	120-121#	120-127	120-127#
	120-131	120-131#	120-146	120-146#	120-150	120-150#	120-156	120-156#	120-160	120-160#	120-166	120-166#	120-180	120-180#
	120-184	120-184#	120-189	120-189#	121-50	121-50#	122-7	122-7#	123-12	123-12#	124-11	124-11#	125-14	125-14#
	126-14	126-14#	126-21	126-21#										
M\$GENB	41-66	41-66#	108-32	108-32#	119-64	119-64#	119-68	119-68#	120-14	120-14#	120-23	120-23#	120-29	120-29#
	120-32	120-32#	120-33	120-33#	120-36	120-36#	120-41	120-41#	120-50	120-50#	120-51	120-51#	120-54	120-54#
	120-57	120-57#	120-59	120-59#	120-83	120-83#	120-121	120-121#	120-127	120-127#	120-131	120-131#	120-146	120-146#
	120-150	120-150#	120-156	120-156#	120-160	120-160#	120-166	120-166#	120-180	120-180#	120-184	120-184#		
M\$GETS	7-27	7-27#	8-27	8-27#	8-29	8-29#	27-17	27-17#	27-22	27-22#	27-28	27-28#	27-32	27-32#
	27-37	27-37#	27-49	27-49#	27-54	27-54#	27-59	27-59#	27-63	27-63#	27-67	27-67#	27-71	27-71#
	27-75	27-75#	27-79	27-79#	27-83	27-83#	67-48	67-48#	81-14	81-14#	82-22	82-22#	95-9	95-9#
	96-3	96-3#	98-28	98-28#	99-156	99-156#	100-14	100-14#	109-15	109-15#	110-18	110-18#	111-30	111-30#
	111-31	111-31#	112-40	112-40#	112-43	112-43#	112-46	112-46#	112-50	112-50#	112-56	112-56#	112-66	112-66#
	113-6	113-6#	117-24	117-24#	118-14	118-14#	120-189	120-189#	120-190	120-190#	122-7	122-7#	124-11	124-11#
	125-15	125-15#												
M\$GETT	99-118#	108-45#	119-117#											
M\$GNGB	5-43#	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77
	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77
	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	6-8	6-8#	7-10	7-10#	8-10	8-10

MSGNIN	8-10#	9-48#	22-21	22-21#	24-12	24-12#	24-22	24-22#	27-14	27-14#	27-19	27-19#	27-24	27-24#
	27-30	27-30#	27-34	27-34#	27-39	27-39#	27-51	27-51#	27-56	27-56#	27-61	27-61#	27-65	27-65#
	27-69	27-69#	27-73	27-73#	27-77	27-77#	27-81	27-81#	67-31	67-31#	81-10	81-10#	82-18	82-18#
	95-5	95-5#	96-1	96-1#	99-38#	99-45	99-45#	100-8	100-8#	101-8	101-8#	110-10	110-10#	111-8
	111-8#	112-16#	121-39#	121-50	121-50#	123-12	123-12#	125-14	125-14#					
	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77
	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77
	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77	5-77
	5-77	5-77	5-77	5-77	5-77	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#	5-77#
	6-8#	6-8#	7-10	7-10#	8-10	8-10#	24-12	24-12	24-12#	24-12#	24-22	24-22	24-22#	24-22#
	27-15	27-15	27-15	27-15	27-15	27-15#	27-15#	27-15#	27-16	27-16	27-16	27-16	27-16	27-16
	27-16	27-16#	27-16#	27-16#	27-16#	27-16#	27-17	27-17#	27-20	27-20	27-20	27-20	27-20	27-20#
	27-20#	27-20#	27-20#	27-21	27-21	27-21	27-21	27-21	27-21	27-21#	27-21#	27-21#	27-21#	27-21#
	27-22	27-22#	27-25	27-25	27-25	27-25	27-25	27-25#	27-25#	27-25#	27-25#	27-26	27-26	27-26
	27-26	27-26	27-26	27-26#	27-26#	27-26#	27-26#	27-26#	27-27	27-27	27-27	27-27	27-27	27-27
	27-27#	27-27#	27-27#	27-27#	27-27#	27-28	27-28#	27-31	27-31	27-31	27-31	27-31	27-31	27-31#
	27-31#	27-31#	27-31#	27-31#	27-32	27-32#	27-35	27-35	27-35	27-35	27-35	27-35#	27-35#	27-35#
	27-35#	27-36	27-36	27-36	27-36	27-36	27-36	27-36#	27-36#	27-36#	27-36#	27-36#	27-37	27-37#
	27-40	27-40	27-40	27-40	27-40	27-40#	27-40#	27-40#	27-40#	27-41	27-41	27-41	27-41	27-41
	27-41#	27-41#	27-41#	27-41#	27-42	27-42	27-42	27-42	27-42	27-42#	27-42#	27-42#	27-42#	27-45
	27-45	27-45	27-45	27-45	27-45	27-45	27-45#	27-45#	27-45#	27-45#	27-45#	27-45#	27-49	27-49#
	27-52	27-52	27-52	27-52	27-52	27-52	27-52#	27-52#	27-52#	27-52#	27-52#	27-53	27-53	27-53
	27-53	27-53	27-53#	27-53#	27-53#	27-53#	27-54	27-54#	27-57	27-57	27-57	27-57	27-57	27-57
	27-57#	27-57#	27-57#	27-57#	27-57#	27-58	27-58	27-58	27-58	27-58	27-58#	27-58#	27-58#	27-58#
	27-59	27-59#	27-62	27-62	27-62	27-62	27-62	27-62	27-62	27-62#	27-62#	27-62#	27-62#	27-62#
	27-62#	27-63	27-63#	27-66	27-66	27-66	27-66	27-66	27-66	27-66	27-66#	27-66#	27-66#	27-66#
	27-66#	27-66#	27-67	27-67#	27-70	27-70	27-70	27-70	27-70	27-70	27-70#	27-70#	27-70#	27-70#
	27-70#	27-71	27-71#	27-74	27-74	27-74	27-74	27-74	27-74	27-74#	27-74#	27-74#	27-74#	27-74#
	27-75	27-75#	27-78	27-78	27-78	27-78	27-78	27-78	27-78#	27-78#	27-78#	27-78#	27-78#	27-79
	27-79#	27-82	27-82	27-82	27-82	27-82	27-82	27-82	27-82#	27-82#	27-82#	27-82#	27-82#	27-82#
	27-83	27-83#	28-7	28-7	28-7	28-7	28-7#	28-7#	28-7#	28-7#	28-7#	28-8	28-8#	32-20
	32-20	32-20	32-20	32-20#	32-20#	32-20#	32-20#	32-20#	32-21	32-21#	34-22	34-22	34-22	34-22
	34-22#	34-22#	34-22#	34-22#	34-22#	34-40	34-40	34-40	34-40	34-40#	34-40#	34-40#	34-40#	34-40#
	34-43	34-43#	34-54	34-54#	35-11	35-11	35-11	35-11	35-11#	35-11#	35-11#	35-11#	35-11#	35-18
	35-18	35-18	35-18	35-18#	35-18#	35-18#	35-18#	35-18#	35-25	35-25	35-25	35-25	35-25#	35-25#
	35-25#	35-25#	35-25#	36-6	36-6	36-6	36-6	36-6#	36-6#	36-6#	36-6#	36-6#	36-6#	41-32
	41-33	41-33#	41-34	41-34	41-34	41-34	41-34	41-34#	41-34#	41-34#	41-34#	41-34#	41-42	41-42
	41-42	41-42	41-42	41-42	41-42	41-42#	41-42#	41-42#	41-42#	41-42#	41-42#	41-42#	41-43	41-43
	41-43	41-43	41-43	41-43#	41-43#	41-43#	41-43#	41-44	41-44	41-44	41-44	41-44	41-44#	41-44#
	41-44#	41-44#	41-45	41-45	41-45	41-45	41-45	41-45#	41-45#	41-45#	41-45#	41-45#	41-53	41-53
	41-53	41-53	41-53	41-53#	41-53#	41-53#	41-53#	41-53#	41-59	41-59	41-59	41-59	41-59	41-59
	41-59#	41-59#	41-59#	41-59#	41-59#	41-62	41-62	41-62	41-62	41-62	41-62#	41-62#	41-62#	41-62#
	41-66	41-66	41-66	41-66	41-66	41-66	41-66	41-66#	41-66#	41-66#	41-66#	41-66#	41-113	41-113
	41-113	41-113	41-113	41-113#	41-113#	41-113#	41-113#	49-28	49-28#	49-28#	49-33	49-33	49-33	49-33
	49-33	49-33	49-33	49-33	49-33#	49-33#	49-33#	49-33#	49-33#	49-33#	49-33#	49-34	49-34	49-34
	49-34	49-34	49-34#	49-34#	49-34#	49-34#	51-27	51-27#	51-27#	51-60	51-60	51-60	51-60	51-60
	51-60#	51-60#	51-60#	51-60#	51-63	51-63	51-63	51-63	51-63	51-63#	51-63#	51-63#	51-63#	52-41
	52-41#	52-41#	52-44	52-44	52-44	52-44	52-44	52-44	52-44#	52-44#	52-44#	52-44#	52-44#	53-21
	53-21	53-21	53-21	53-21	53-21	53-21	53-21	53-21#	53-21#	53-21#	53-21#	53-21#	53-21#	53-21#
	55-18	55-18	55-18	55-18	55-18	55-18#	55-18#	55-18#	55-18#	56-6	56-6	56-6	56-6	56-6
	56-6	56-6#	56-6#	56-6#	56-6#	56-6#	56-14	56-14	56-14	56-14	56-14	56-14	56-14#	56-14#
	56-14#	56-14#	56-14#	56-43	56-43	56-43	56-43	56-43	56-43#	56-43#	56-43#	56-43#	56-56	56-56
	56-56	56-56	56-56	56-56#	56-56#	56-56#	56-56#	56-64	56-64	56-64	56-64	56-64	56-64	56-64#
	56-64#	56-64#	56-64#	56-64#	58-13	58-13	58-13	58-13	58-13	58-13#	58-13#	58-13#	58-13#	58-20

58-20	58-20	58-20	58-20	58-20#	58-20#	58-20#	58-20#	58-20#	58-20#	59-24	59-24	59-24	59-24#	59-24#
59-24#	59-24#	59-24#	61-59	61-59	61-59	61-59	61-59	61-59	61-59	61-59	61-59#	61-59#	61-59#	61-59#
61-69	61-69	61-69	61-69	61-69	61-69	61-69	61-69	61-69	61-69	61-69	61-69#	61-69#	61-69#	61-69#
64-39	64-39	64-39#	64-39#	64-39#	64-39#	64-39#	64-43	64-43	64-43	64-43	64-43	64-43	64-43#	64-43#
64-43#	65-4	65-4	65-4	65-4	65-4	65-4	65-4	65-4#	65-4#	65-4#	65-4#	65-4#	65-4#	65-13
65-13	65-13	65-13	65-13	65-13#	65-13#	65-13#	65-13#	65-13#	65-13#	65-13#	65-13#	65-13#	65-13#	65-13
65-30	65-30#	65-30#	65-30#	65-30#	65-30#	65-30#	67-27	67-27#	67-27#	67-27#	67-27#	67-27#	67-27#	67-27#
67-34	67-34	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#	67-34#
67-37	67-37#	67-37#	67-37#	67-37#	67-37#	67-37#	67-48	67-48#	67-48#	68-32	68-32	68-32	68-32	68-32
68-32#	68-32#	68-32#	68-32#	68-32#	68-32#	68-32#	71-3	71-3	71-3	71-3	71-3#	71-3#	71-3#	71-3#
71-3#	75-28	75-28#	75-36	75-36	75-36	75-36	75-36#	75-36#	75-36#	75-36#	75-36#	75-36#	75-36#	75-44
75-44	75-44	75-44#	75-44#	75-44#	75-44#	75-44#	77-17	77-17	77-17	77-17	77-17	77-17	77-17	77-17#
77-17#	77-17#	77-17#	77-17#	77-17#	77-17#	77-36	77-36	77-36#	77-36#	81-14	81-14#	82-22	82-22#	85-18
85-18	85-18	85-18	85-18#	85-18#	85-18#	85-18#	85-18#	85-18#	85-18#	85-37	85-37	85-37	85-37#	85-37#
85-37#	85-37#	85-37#	86-13	86-13#	86-24	86-24	86-24	86-24	86-24	86-24	86-24	86-24#	86-24#	86-24#
86-24#	86-24#	86-24#	86-27	86-27#	86-27#	86-27#	86-30	86-30	86-30	86-30	86-30	86-30#	86-30#	86-30#
86-30#	86-30#	87-44	87-44	87-44#	87-44#	87-44#	87-44#	87-44#	87-44#	87-44#	87-44#	88-27	88-27#	88-36
88-36	88-36	88-36	88-36#	88-36#	88-36#	88-36#	88-36#	88-36#	88-36#	89-5	89-5	89-5	89-5#	89-5#
89-5#	89-5#	89-5#	92-12	92-12#	93-21	93-21	93-21#	93-21#	93-21#	93-24	93-24#	93-26	93-26#	93-26#
93-26#	93-31	93-31#	93-31#	93-41	93-41#	93-41#	93-76	93-76#	93-76#	93-83	93-83#	93-83#	93-83#	93-89
93-87#	93-89#	93-99	93-99#	93-99#	93-102	93-102#	94-1	94-1	94-1	94-1	94-1	94-1#	94-1#	94-1#
94-1#	94-1#	94-1#	94-2	94-2#	94-3	94-3	94-3	94-3	94-3	94-3#	94-3#	94-3#	94-3#	94-3#
94-4	94-4#	95-9	95-9#	96-3	96-3#	97-10	97-10#	97-11	97-11#	98-23	98-23	98-23	98-23	98-23
98-23	98-23	98-23	98-23#	98-23#	98-23#	98-23#	98-23#	98-23#	98-23#	98-23#	98-23#	98-24	98-24	98-24
98-24	98-24	98-24#	98-24#	98-24#	98-24#	99-59	99-59	99-59	99-59	99-59	99-59	99-59	99-59#	99-59#
99-59#	99-59#	99-59#	99-70	99-70	99-70	99-70	99-70	99-70	99-70	99-70	99-70	99-70#	99-70#	99-70#
99-70#	99-70#	99-70#	99-70#	99-71	99-71	99-71	99-71	99-71	99-71	99-71#	99-71#	99-71#	99-71#	99-74
99-74	99-74	99-74	99-74#	99-74#	99-74#	99-74#	99-74#	99-74#	99-74#	99-75	99-75	99-75	99-75#	99-75#
99-75#	99-75#	99-75#	99-88	99-88	99-88	99-88	99-88	99-88	99-88	99-88	99-88#	99-88#	99-88#	99-88#
99-88#	99-88#	99-107	99-107	99-107	99-107	99-107	99-107	99-107	99-107	99-107	99-107	99-107#	99-107#	99-107#
99-107#	99-107#	99-107#	99-107#	99-107#	99-108	99-108	99-108	99-108	99-108	99-108	99-108	99-108	99-108	99-108#
99-108#	99-108#	99-108#	99-108#	99-108#	99-108#	99-118	99-118	99-118	99-118	99-118#	99-118#	99-156	99-156#	101-63
101-63#	101-63#	101-64	101-64#	101-66	101-66	101-66#	101-66#	101-67	101-67#	101-69	101-69	101-69#	101-69#	101-69#
101-70	101-70#	101-73	101-73#	101-79	101-79#	101-87	101-87	101-87	101-87	101-87	101-87	101-87	101-87	101-87
101-87#	101-88	101-83#	101-91	101-91	101-91#	101-91#	101-92	101-92#	101-92#	101-114	101-114	101-114#	101-114#	101-114#
101-115	101-115#	101-122	101-122	101-122	101-122	101-122#	101-122#	101-122#	101-122#	101-122#	101-122#	101-123	101-123#	102-1
102-1#	102-3	102-3	102-3#	102-3#	102-13	102-13	102-13	102-13#	102-13#	102-13#	102-14	102-14#	102-15	102-15
102-15#	102-15#	102-15#	102-16	102-16#	102-18	102-18	102-18	102-18	102-18	102-18	102-18#	102-18#	102-18#	102-18#
102-24	102-24	102-24	102-24	102-24	102-24	102-24#	102-24#	102-24#	102-24#	102-24#	102-24#	102-24#	103-10	103-10
103-10#	103-10#	103-10#	103-11	103-11#	106-5	106-5	106-5#	106-5#	106-5#	106-5#	106-6	106-6#	108-12	108-12
108-12	108-12	108-12	108-12#	108-12#	108-12#	108-12#	108-21	108-21	108-21	108-21	108-21	108-21	108-21	108-21
108-21	108-21#	108-21#	108-21#	108-21#	108-21#	108-21#	108-30	108-30#	108-30#	108-31	108-31#	108-32	108-32	108-32
108-32	108-32	108-32	108-32	108-32#	108-32#	108-32#	108-35	108-35#	108-35#	108-42	108-42	108-42#	108-42#	108-42#
108-45	108-45	108-45#	108-45#	109-2	109-2	109-2	109-2	109-2#	109-2#	109-2#	109-2#	109-2#	109-2#	109-3
109-3#	109-6	109-6	109-6	109-6	109-6#	109-6#	109-6#	109-6#	109-6#	109-6#	109-7	109-7#	109-11	109-11
109-11	109-11	109-11#	109-11#	109-11#	109-11#	109-11#	109-12	109-12#	109-15	109-15#	110-18	110-18#	111-30	111-30
111-30#	112-28	112-28#	112-30	112-30	112-30	112-30	112-30	112-30	112-30#	112-30#	112-30#	112-30#	112-30#	112-30#
112-30#	112-34	112-34	112-34#	112-34#	112-37	112-37	112-37	112-37	112-37#	112-37#	112-37#	112-37#	112-37#	112-37#
112-40	112-40#	112-41	112-41#	112-43	112-43#	112-44	112-44#	112-46	112-46#	112-47	112-47#	112-50	112-50#	112-50#
112-51	112-51#	112-56	112-56#	112-57	112-57#	112-66	112-66#	113-6	113-6#	114-17	114-17	114-17	114-17	114-17
114-17#	114-17#	114-17#	114-17#	114-27	114-27	114-27	114-27	114-27	114-27#	114-27#	114-27#	114-27#	114-27#	114-27#
115-11	115-11#	115-19	115-19	115-19	115-19	115-19#	115-19#	115-19#	115-19#	115-19#	115-19#	116-16	116-16	116-16
116-16	116-16	116-16	116-16	116-16#	116-16#	116-16#	116-16#	116-16#	116-16#	116-16#	116-17	116-17	116-17	116-17
116-17	116-17	116-17#	116-17#	116-17#	116-17#	116-17#	116-17#	116-17#	116-17#	116-18	116-18	116-18#	116-18#	116-18#
116-41	116-41	116-41	116-41	116-41#	116-41#	116-41#	116-41#	116-41#	116-41#	116-44	116-44	116-44#	116-44#	116-55
116-55	116-55#	116-55#	116-61	116-61	116-61	116-61	116-61	116-61#	116-61#	116-61#	116-61#	116-61#	116-63	116-63
116-63#	116-63#	116-67	116-67	116-67	116-67	116-67#	116-67#	116-67#	116-67#	116-67#	116-67#	116-69	116-69	116-69

	116-69	116-69#	118-14	118-14#	119-12	119-12#	119-12	119-12#	119-12	119-12#	119-12	119-12#	119-19
	117-24	117-24#	119-19	119-19#	119-12	119-12#	119-12	119-12#	119-24	119-24#	119-24	119-24#	119-24#
	119-19	119-19#	119-24	119-24#	119-19#	119-19#	119-19#	119-19#	119-64	119-64#	119-64	119-64#	119-64#
	119-24#	119-24#	119-36	119-36#	119-37	119-37#	119-37#	119-37#	119-64	119-64#	119-64	119-64#	119-64#
	119-64	119-64#	119-64#	119-64#	119-64#	119-68	119-68#	119-68#	119-68	119-68#	119-68	119-68#	119-68#
	119-68#	119-68#	119-68#	119-76	119-76#	119-76	119-76#	119-76#	119-76	119-76#	119-76#	119-76#	119-113
	119-116	119-116#	119-117	119-117#	119-117#	119-117#	119-117#	119-117#	120-12	120-12#	120-12	120-12#	120-12
	120-12#	120-12#	120-12#	120-12#	120-12#	120-12#	120-12#	120-12#	120-14	120-14#	120-14	120-14#	120-14
	120-14	120-14#	120-14#	120-14#	120-14#	120-23	120-23#	120-23#	120-23	120-23#	120-23	120-23#	120-23#
	120-23#	120-23#	120-23#	120-29	120-29#	120-29	120-29#	120-29#	120-29	120-29#	120-29#	120-29#	120-32
	120-32	120-32#	120-32	120-32	120-32#	120-32#	120-32#	120-32#	120-32#	120-33	120-33#	120-33#	120-33#
	120-33	120-33#	120-33#	120-33#	120-33#	120-36	120-36#	120-36#	120-36	120-36#	120-36#	120-36#	120-36#
	120-36#	120-41	120-41	120-41	120-41	120-41	120-41#	120-41#	120-41	120-41#	120-41#	120-41#	120-50
	120-50	120-50	120-50	120-50	120-50#	120-50#	120-50#	120-50#	120-50#	120-51	120-51#	120-51#	120-51
	120-51	120-51#	120-51#	120-51#	120-51#	120-54	120-54#	120-54#	120-54	120-54#	120-54#	120-54#	120-54#
	120-54#	120-57	120-57	120-57	120-57#	120-57	120-57#	120-57#	120-57#	120-57#	120-57#	120-57#	120-59
	120-59	120-59	120-59	120-59#	120-59#	120-59#	120-59#	120-59#	120-77	120-77#	120-77#	120-77#	120-77#
	120-77#	120-77#	120-78	120-78#	120-78#	120-78#	120-78#	120-78#	120-78#	120-78#	120-78#	120-78#	120-79
	120-79	120-79	120-79#	120-79#	120-79#	120-79#	120-79#	120-79#	120-80	120-80#	120-80#	120-80#	120-80#
	120-80#	120-81	120-81	120-81#	120-81#	120-81#	120-81#	120-81#	120-81#	120-81#	120-82	120-82#	120-82#
	120-82	120-82#	120-82#	120-82#	120-82#	120-83	120-83#	120-83#	120-83	120-83#	120-83	120-83#	120-83#
	120-83#	120-83#	120-83#	120-121	120-121#	120-121	120-121#	120-121#	120-121	120-121#	120-121#	120-121#	120-121#
	120-121#	120-127	120-127	120-127	120-127#	120-127	120-127#	120-127#	120-127	120-127#	120-127#	120-127#	120-131
	120-131	120-131	120-131	120-131	120-131#	120-131	120-131#	120-131#	120-131#	120-131#	120-131#	120-131#	120-146
	120-146	120-146	120-146	120-146	120-146#	120-146#	120-146#	120-146#	120-150	120-150#	120-150#	120-150#	120-150
	120-150	120-150	120-150	120-150#	120-150#	120-150#	120-150#	120-150#	120-156	120-156#	120-156#	120-156#	120-156
	120-156	120-156#	120-156#	120-156#	120-156#	120-160	120-160#	120-160#	120-160	120-160#	120-160#	120-160#	120-160#
	120-160#	120-160#	120-160#	120-166	120-166#	120-166	120-166#	120-166#	120-166	120-166#	120-166#	120-166#	120-180
	120-180	120-180	120-180	120-180	120-180#	120-180	120-180#	120-180#	120-180#	120-180#	120-180#	120-184	120-184
	120-184	120-184	120-184	120-184	120-184#	120-184#	120-184#	120-184#	120-184#	120-189	120-189#	121-50	121-50#
	122-1	122-1	122-1	122-1#	122-2	122-2	122-2#	122-2#	122-2#	122-3	122-3#	122-3	122-3
	122-3#	122-4	122-4	122-4#	122-4	122-4#	122-4#	122-4#	122-5	122-5#	122-5	122-5#	122-6
	122-6	122-6	122-6#	122-7	122-7#	123-12	123-12#	124-1	124-1#	124-1	124-1#	124-2	124-2#
	124-2#	124-3	124-3	124-3	124-3#	124-3	124-3#	124-4	124-4#	124-4	124-4#	124-4#	124-5
	124-5	124-5	124-5#	124-6	124-6#	124-6	124-6#	124-7	124-7#	124-7	124-7#	124-11	125-14
	125-14	125-14	125-14#	126-14	126-14#	126-14	126-14#	126-14#	126-14#	126-14#	126-14#	126-14#	126-14#
MSGNLS	41-66	41-66#	108-32	108-32#	119-64	119-64#	119-68	119-68#	120-14	120-14#	120-23	120-23#	120-29
	120-32	120-32#	120-33	120-33#	120-36	120-36#	120-41	120-41#	120-50	120-50#	120-51	120-51#	120-54
	120-57	120-57#	120-59	120-59#	120-83	120-83#	120-121	120-121#	120-127	120-127#	120-131	120-131#	120-146
	120-150	120-150#	120-156	120-156#	120-160	120-160#	120-166	120-166#	120-180	120-180#	120-184	120-184#	120-184#
MSGNSU	112-28	112-28#	112-41	112-41#	112-44	112-44#	112-47	112-47#	112-51	112-51#	112-57	112-57#	112-57#
MSGNTA	7-27	7-27#	8-27	8-27#	27-17	27-17#	27-22	27-22#	27-28	27-28#	27-32	27-32#	27-37
	27-49	27-49#	27-54	27-54#	27-59	27-59#	27-63	27-63#	27-67	27-67#	27-71	27-71#	27-75
	27-79	27-79#	27-83	27-83#	67-48	67-48#	81-14	81-14#	82-22	82-22#	95-9	95-9#	96-3
	99-156	99-156#	109-15	109-15#	110-18	110-18#	111-30	111-30#	112-40	112-40#	112-43	112-43#	112-46
	112-50	112-50#	112-56	112-56#	112-66	112-66#	113-6	113-6#	117-24	117-24#	118-14	118-14#	120-189
	122-7	122-7#	124-11	124-11#	126-14	126-14#	126-21	126-21#	126-21#	126-21#	126-21#	126-21#	126-21#
MSGNTE	112-18	112-18#	117-3	117-3#	118-3	118-3#	119-3	119-3#	119-3#	119-3#	119-3#	119-3#	119-3#
M\$HAPT	5-77	5-77#											
M\$HNAP	5-77	5-77#											
M\$INCR	5-43	5-43#	7-10	7-10	7-10#	7-10#	8-10	8-10	8-10#	8-10#	9-48	9-48#	27-14
	27-14#	27-14#	27-15#	27-16#	27-17#	27-19	27-19	27-19#	27-19#	27-20#	27-21#	27-22#	27-24
	27-24#	27-24#	27-25#	27-26#	27-27#	27-28#	27-30	27-30	27-30#	27-30#	27-31#	27-32#	27-34
	27-34#	27-34#	27-35#	27-36#	27-37#	27-39	27-39	27-39#	27-39#	27-40#	27-41#	27-42#	27-49#
	27-51	27-51	27-51#	27-51#	27-52#	27-53#	27-54#	27-56	27-56	27-56#	27-56#	27-57#	27-58#
	27-61	27-61	27-61#	27-61#	27-62#	27-63#	27-65	27-65	27-65#	27-65#	27-66#	27-67#	27-69
	27-69#	27-69#	27-70#	27-71#	27-73	27-73	27-73#	27-73#	27-74#	27-75#	27-77	27-77#	27-77#

	27-78#	27-79#	27-81	27-81	27-81#	27-81#	27-82#	27-83#	28-7#	28-8#	32-20#	32-21#	34-22#	34-40#
	34-43#	34-54#	35-11#	35-18#	35-25#	36-6#	41-32#	41-34#	41-42#	41-43#	41-44#	41-45#	41-53#	41-59#
	41-62#	41-66	41-66#	41-66#	41-113#	49-28#	49-33#	49-34#	51-27#	51-60#	51-63#	52-41#	52-44#	53-21#
	55-18#	56-6#	56-14#	56-43#	56-56#	56-64#	58-13#	58-20#	59-24#	61-59#	61-69#	64-39#	64-43#	65-4#
	65-13#	65-30#	67-27#	67-31	67-31	67-31#	67-31#	67-34#	67-37#	67-48#	68-32#	71-3#	75-28#	75-36#
	75-44#	77-17#	77-36#	81-10	81-10	81-10#	81-10#	82-18	82-18	82-18#	82-18#	85-18#	85-37#	86-13#
	86-24#	86-27#	86-30#	87-44#	88-27#	88-36#	89-5#	92-12#	93-21#	93-24#	93-26#	93-31#	93-41#	93-76#
	93-83#	93-89#	93-99#	93-102#	94-1#	94-2#	94-3#	94-4#	95-5	95-5	95-5#	95-5#	96-1	96-1
	96-1#	96-1#	97-10#	97-11#	98-23#	98-24#	99-38	99-38#	99-45	99-45	99-45#	99-45#	99-59#	99-70#
	99-71#	99-74#	99-75#	99-88#	99-107#	99-108#	99-156#	100-8	100-8	100-8#	100-8#	101-8	101-8	101-8#
	101-8#	101-63#	101-66#	101-69#	101-73#	101-79#	101-88#	101-91#	101-114#	101-122#	101-123#	102-1#	102-3#	102-13#
	102-15#	102-18#	102-24#	103-10#	106-5#	108-12#	108-21#	108-30#	108-32	108-32#	108-32#	108-35#	108-42#	108-45#
	109-2#	109-3#	109-6#	109-7#	109-11#	109-12#	109-15#	110-10	110-10	110-10#	110-10#	110-18#	111-8	111-8
	111-8#	111-8#	111-30#	112-16	112-16#	112-18	112-18	112-18	112-18#	112-18#	112-18#	112-28	112-28	112-28
	112-28#	112-28#	112-28#	112-30#	112-34#	112-37#	112-40#	112-41	112-41	112-41	112-41#	112-41#	112-41#	112-43#
	112-44	112-44	112-44	112-44#	112-44#	112-44#	112-46#	112-47	112-47	112-47	112-47#	112-47#	112-47#	112-50#
	112-51	112-51	112-51	112-51#	112-51#	112-51#	112-56#	112-57	112-57	112-57	112-57#	112-57#	112-57#	112-66#
	113-6#	114-17#	114-27#	115-11#	115-19#	116-16#	116-17#	116-18#	116-33#	116-41#	116-44#	116-55#	116-61#	116-63#
	116-67#	116-69#	116-72#	116-77#	117-3	117-3	117-3	117-3#	117-3#	117-3#	117-24#	118-3	118-3	118-3
	118-3#	118-3#	118-3#	118-14#	119-3	119-3	119-3	119-3#	119-3#	119-3#	119-12#	119-19#	119-24#	119-36#
	119-64	119-64#	119-64#	119-68	119-68#	119-68#	119-76#	119-113#	119-116#	119-117#	120-12#	120-14	120-14#	120-14#
	120-23	120-23#	120-23#	120-29	120-29#	120-29#	120-32	120-32#	120-32#	120-33	120-33#	120-33#	120-36	120-36#
	120-36#	120-41	120-41#	120-41#	120-50	120-50#	120-50#	120-51	120-51#	120-51#	120-54	120-54#	120-54#	120-57
	120-57#	120-57#	120-59	120-59#	120-59#	120-77#	120-78#	120-79#	120-80#	120-81#	120-82#	120-83	120-83#	120-83#
	120-121	120-121#	120-121#	120-127	120-127#	120-127#	120-131	120-131#	120-131#	120-146	120-146#	120-146#	120-150	120-150#
	120-150#	120-156	120-156#	120-156#	120-160	120-160#	120-160#	120-166	120-166#	120-166#	120-180	120-180#	120-180#	120-184
	120-184#	120-184#	120-189#	121-39	121-39#	121-50	121-50	121-50#	121-50#	123-12	123-12	123-12#	123-12#	126-13
	126-13#	126-14	126-14	126-14	126-14#									
M\$LDRO	77-36	77-36#	86-27	86-27#	93-21	93-21#	101-63	101-63#	101-66	101-66#	101-69	101-69#	101-91	101-91#
	101-114	101-114#	102-13	102-13#	102-15	102-15#	103-10	103-10#	106-5	106-5#	108-42	108-42#	112-34	112-34#
	116-18	116-18#	116-44	116-44#	116-55	116-55#	116-63	116-63#	116-72	116-72#	116-77	116-77#		
M\$MCHI	5-16	5-16#												
M\$MCLO	5-16	5-16#												
M\$POP	7-27	7-27#	8-27	8-27#	8-29	8-29#	27-17	27-17#	27-22	27-22#	27-28	27-28#	27-32	27-32#
	27-37	27-37#	27-43	27-49#	27-54	27-54#	27-59	27-59#	27-63	27-63#	27-67	27-67#	27-71	27-71#
	27-75	27-75#	27-79	27-79#	27-83	27-83#	67-48	67-48#	81-14	81-14#	82-22	82-22#	95-9	95-9#
	96-3	96-3#	98-28	98-28#	99-156	99-156#	100-14	100-14#	109-15	109-15#	110-18	110-18#	111-30	111-30#
	111-31	111-31#	112-40	112-40#	112-43	112-43#	112-46	112-46#	112-50	112-50#	112-56	112-56#	112-66	112-66#
	113-6	113-6#	117-24	117-24#	118-14	118-14#	120-189	120-189#	120-190	120-190#	122-7	122-7#	124-11	124-11#
	125-15	125-15#												
M\$PRIN	27-15	27-15#	27-16	27-16#	27-20	27-20#	27-21	27-21#	27-25	27-25#	27-26	27-26#	27-27	27-27#
	27-31	27-31#	27-35	27-35#	27-36	27-36#	27-40	27-40#	27-41	27-41#	27-42	27-42#	27-45	27-45#
	27-52	27-52#	27-53	27-53#	27-57	27-57#	27-58	27-58#	27-62	27-62#	27-66	27-66#	27-70	27-70#
	27-74	27-74#	27-78	27-78#	27-82	27-82#	41-34	41-34#	41-42	41-42#	41-43	41-43#	41-44	41-44#
	41-45	41-45#	41-53	41-53#	41-59	41-59#	41-62	41-62#	41-113	41-113#	49-33	49-33#	49-34	49-34#
	51-60	51-60#	51-63	51-63#	52-44	52-44#	53-21	53-21#	55-18	55-18#	56-6	56-6#	56-14	56-14#
	56-43	56-43#	56-56	56-56#	56-64	56-64#	58-13	58-13#	58-20	58-20#	61-59	61-59#	61-69	61-69#
	64-39	64-39#	64-43	64-43#	65-4	65-4#	65-13	65-13#	65-30	65-30#	67-34	67-34#	67-37	67-37#
	98-23	98-23#	98-24	98-24#	99-59	99-59#	99-70	99-70#	99-71	99-71#	99-74	99-74#	99-75	99-75#
	99-88	99-88#	99-107	99-107#	99-108	99-108#	102-18	102-18#	108-12	108-12#	108-21	108-21#	116-16	116-16#
	116-69	116-69#	119-12	119-12#	119-19	119-19#	119-24	119-24#	119-76	119-76#	120-12	120-12#	120-77	120-77#
	120-78	120-78#	120-79	120-79#	120-80	120-80#	120-81	120-81#	120-82	120-82#	120-82	120-82#		
M\$PUSH	5-43	5-43#	7-10	7-10#	8-10	8-10#	9-48	9-48#	27-14	27-14#	27-19	27-19#	27-24	27-24#
	27-30	27-30#	27-34	27-34#	27-39	27-39#	27-51	27-51#	27-56	27-56#	27-61	27-61#	27-65	27-65#
	27-69	27-69#	27-73	27-73#	27-77	27-77#	27-81	27-81#	67-31	67-31#	81-10	81-10#	82-18	82-18#
	95-5	95-5#	96-1	96-1#	99-38	99-38#	99-45	99-45#	100-8	100-8#	101-8	101-8#	110-10	110-10#
	111-8	111-8#	112-16	112-16#	112-18	112-18#	112-28	112-28#	112-41	112-41#	112-44	112-44#	112-47	112-47#

	112-51	112-51#	112-57	112-57#	117-3	117-3#	118-3	118-3#	119-3	119-3#	121-39	121-39#	121-50	121-50#
MSPUT	112-51	112-51#	112-57	112-57#	117-3	117-3#	118-3	118-3#	119-3	119-3#	121-39	121-39#	121-50	121-50#
	123-12	123-12#												
	27-15	27-15	27-15#	27-16	27-16	27-16	27-16#	27-20	27-20	27-20#	27-21	27-21	27-21	27-21#
	27-25	27-25	27-25#	27-26	27-26	27-26	27-26#	27-27	27-27	27-27#	27-27#	27-31	27-31	27-31
	27-31#	27-35	27-35#	27-35#	27-36	27-36	27-36#	27-36#	27-40	27-40	27-40#	27-41	27-41	27-41#
	27-42	27-42	27-42#	27-45	27-45	27-45	27-45#	27-45#	27-52	27-52	27-52	27-52#	27-53	27-53
	27-53#	27-57	27-57	27-57#	27-57#	27-58	27-58	27-58#	27-62	27-62	27-62	27-62	27-62#	27-66
	27-66	27-66	27-66	27-66#	27-70	27-70	27-70	27-70#	27-74	27-74	27-74	27-74#	27-78	27-78
	27-78	27-78#	27-82	27-82	27-82	27-82	27-82#	41-34	41-34	41-34#	41-42	41-42	41-42	41-42
	41-42	41-42#	41-43	41-43	41-43#	41-44	41-44	41-44#	41-45	41-45	41-45#	41-53	41-53	41-53
	41-53#	41-59	41-59	41-59	41-59#	41-62	41-62	41-62#	41-113	41-113	41-113#	49-33	49-33	49-33
	49-33	49-33	49-33#	49-34	49-34	49-34#	51-60	51-60	51-60#	51-63	51-63	51-63#	52-44	52-44
	52-44	52-44#	53-21	53-21	53-21	53-21	53-21#	53-21#	55-18	55-18	55-18#	56-6	56-6	56-6
	56-6#	56-14	56-14	56-14	56-14#	56-43	56-43	56-43#	56-56	56-56	56-56#	56-64	56-64	56-64
	56-64#	58-13	58-13	58-13#	58-20	58-20	58-20#	61-59	61-59	61-59	61-59#	61-69	61-69	61-69
	61-69#	64-39	64-39	64-39#	64-43	64-43	64-43#	65-4	65-4	65-4	65-4#	65-13	65-13	65-13
	65-13#	65-30	65-30	65-30#	67-34	67-34	67-34	67-34	67-34	67-34	67-34#	67-37	67-37	67-37
	67-37#	68-32	68-32	68-32	68-32	68-32#	77-17	77-17	77-17	77-17	77-17#	86-24	86-24	86-24
	86-24	86-24#	98-23	98-23	98-23	98-23	98-23	98-23#	98-24	98-24	98-24#	99-59	99-59	99-59
	99-59#	99-70	99-70	99-70	99-70	99-70	99-70#	99-71	99-71	99-71#	99-74	99-74	99-74#	99-75
	99-75	99-75#	99-88	99-88	99-88	99-88	99-88#	99-107	99-107	99-107	99-107	99-107	99-107	99-107#
	99-108	99-108	99-108	99-108	99-108	99-108#	102-18	102-18	102-18#	102-24	102-24	102-24	102-24	102-24#
	108-12	108-12	108-12#	108-21	108-21	108-21	108-21	108-21	108-21#	112-30	112-30	112-30	112-30	112-30#
	116-16	116-16	116-16	116-16	116-16#	116-17	116-17	116-17	116-17	116-17#	116-69	116-69	116-69#	119-12
	119-12	119-12#	119-19	119-19	119-19#	119-24	119-24	119-24#	119-76	119-76	119-76#	120-12	120-12	120-12
	120-12	120-12	120-12#	120-77	120-77	120-77#	120-78	120-78	120-78	120-79	120-79	120-79#	120-80	120-80
MSPUT1	120-80#	120-81	120-81	120-81#	120-82	120-82	120-82#							
	27-15	27-15	27-15#	27-15#	27-16	27-16	27-16	27-16#	27-16#	27-16#	27-20	27-20	27-20#	27-20#
	27-21	27-21	27-21	27-21#	27-21#	27-21#	27-25	27-25	27-25#	27-25#	27-26	27-26	27-26	27-26#
	27-26#	27-26#	27-27	27-27	27-27	27-27#	27-27#	27-27#	27-31	27-31	27-31	27-31#	27-31#	27-31#
	27-35	27-35	27-35#	27-35#	27-36	27-36	27-36	27-36#	27-36#	27-36#	27-40	27-40	27-40#	27-40#
	27-41	27-41	27-41#	27-41#	27-42	27-42	27-42#	27-42#	27-45	27-45	27-45	27-45	27-45#	27-45#
	27-45#	27-45#	27-52	27-52	27-52	27-52#	27-52#	27-52#	27-53	27-53	27-53#	27-53#	27-57	27-57
	27-57	27-57#	27-57#	27-57#	27-58	27-58	27-58#	27-58#	27-62	27-62	27-62	27-62	27-62#	27-62#
	27-62#	27-62#	27-65	27-66	27-66	27-66	27-66#	27-66#	27-66#	27-66#	27-70	27-70	27-70	27-70#
	27-70#	27-70#	27-74	27-74	27-74	27-74#	27-74#	27-74#	27-78	27-78	27-78	27-78#	27-78#	27-78#
	27-82	27-82	27-82	27-82	27-82#	27-82#	27-82#	27-82#	41-34	41-34	41-34#	41-34#	41-42	41-42
	41-42	41-42	41-42	41-42#	41-42#	41-42#	41-42#	41-42#	41-43	41-43	41-43#	41-43#	41-44	41-44
	41-44#	41-44#	41-45	41-45	41-45#	41-45#	41-53	41-53	41-53	41-53#	41-53#	41-53#	41-59	41-59
	41-59	41-59#	41-59#	41-59#	41-62	41-62	41-62#	41-62#	41-113	41-113	41-113#	41-113#	49-33	49-33
	49-33	49-33	49-33	49-33#	49-33#	49-33#	49-33#	49-33#	49-34	49-34	49-34#	49-34#	51-60	51-60
	51-60#	51-60#	51-63	51-63	51-63#	51-63#	52-44	52-44	52-44	52-44#	52-44#	52-44#	53-21	53-21
	53-21	53-21	53-21	53-21#	53-21#	53-21#	53-21#	53-21#	55-18	55-18	55-18#	55-18#	56-6	56-6
	56-6	56-6#	56-6#	56-6#	56-14	56-14	56-14	56-14#	56-14#	56-14#	56-43	56-43	56-43#	56-43#
	56-56	56-56	56-56#	56-56#	56-64	56-64	56-64	56-64#	56-64#	56-64#	58-13	58-13	58-13#	58-13#
	58-20	58-20	58-20#	58-20#	61-59	61-59	61-59	61-59#	61-59#	61-59#	61-69	61-69	61-69	61-69#
	61-69#	61-69#	64-39	64-39	64-39#	64-39#	64-43	64-43	64-43#	64-43#	65-4	65-4	65-4	65-4#
	65-4#	65-4#	65-13	65-13	65-13	65-13#	65-13#	65-13#	65-30	65-30	65-30	65-30#	65-30#	65-30#
	67-34	67-34	67-34	67-34	67-34	67-34#	67-34#	67-34#	67-34#	67-34#	67-37	67-37	67-37	67-37#
	67-37#	67-37#	68-32	68-32	68-32	68-32	68-32#	68-32#	68-32#	68-32#	77-17	77-17	77-17	77-17
	77-17#	77-17#	77-17#	77-17#	86-24	86-24	86-24	86-24	86-24#	86-24#	86-24#	86-24#	88-23	88-23
	98-23	98-23	98-23	98-23#	98-23#	98-23#	98-23#	98-23#	98-24	98-24	98-24#	98-24#	99-59	99-59
	99-59	99-59#	99-59#	99-59#	99-70	99-70	99-70	99-70	99-70	99-70#	99-70#	99-70#	99-70#	99-70#
	99-71	99-71	99-71#	99-71#	99-74	99-74	99-74	99-74#	99-75	99-75	99-75#	99-75#	99-88	99-88
	99-88	99-88	99-88#	99-88#	99-88#	99-88#	99-107	99-107	99-107	99-107	99-107	99-107	99-107#	99-107#
	99-107#	99-107#	99-107#	99-107#	99-108	99-108	99-108	99-108	99-108	99-108#	99-108#	99-108#	99-108#	99-108#
	102-18	102-18	102-18#	102-18#	102-24	102-24	102-24	102-24	102-24	102-24#	102-24#	102-24#	108-12	108-12

	108-12#	108-12#	108-21	108-21	108-21	108-21	108-21	108-21#	108-21#	108-21#	108-21#	108-21#	112-30	112-30
	112-30	112-30	112-30#	112-30#	112-30#	112-30#	116-16	116-16	116-16	116-16	116-16#	116-16#	116-16#	116-16#
	116-17	116-17	116-17	116-17	116-17#	116-17#	116-17#	116-17#	116-69	116-69	116-69#	116-69#	119-12	119-12
	119-12#	119-12#	119-19	119-19	119-19#	119-19#	119-24	119-24	119-24#	119-24#	119-76	119-76	119-76#	119-76#
	120-12	120-12	120-12	120-12	120-12	120-12#	120-12#	120-12#	120-12#	120-12#	120-77	120-77	120-77#	120-77#
	120-78	120-78	120-78#	120-78#	120-79	120-79	120-79#	120-79#	120-80	120-80	120-80#	120-80#	120-81	120-81
	120-81#	120-81#	120-82	120-82	120-82#	120-82#								
M\$RADI	41-66	41-66#	108-32	108-32#	119-64	119-64#	119-68	119-68#	120-14	120-14#	120-23	120-23#	120-29	120-29#
	120-32	120-32#	120-33	120-33#	120-36	120-36#	120-41	120-41#	120-50	120-50#	120-51	120-51#	120-54	120-54#
	120-57	120-57#	120-59	120-59#	120-83	120-83#	120-121	120-121#	120-127	120-127#	120-131	120-131#	120-146	120-146#
	120-150	120-150#	120-156	120-156#	120-160	120-160#	120-166	120-166#	120-180	120-180#	120-184	120-184#	122-1	122-1#
	122-2	122-2#	122-3	122-3#	122-4	122-4#	122-5	122-5#	122-6	122-6#	124-1	124-1#	124-3	124-3#
	124-4	124-4#	124-5	124-5#	124-6	124-6#	124-7	124-7#						
M\$RBRO	93-26	93-26#	93-31	93-31#	93-41	93-41#	93-76	93-76#	93-83	93-83#	93-89	93-89#	93-99	93-99#
	93-102	93-102#												
M\$RNRO	49-28	49-28#	51-27	51-27#	52-41	52-41#	101-114	101-114#	102-3	102-3#	102-13	102-13#	102-15	102-15#
	103-10	103-10#	106-5	106-5#										
M\$SETS	5-43	5-43#	7-10	7-10#	8-10	8-10#	9-48	9-48#	27-14	27-14#	27-19	27-19#	27-24	27-24#
	27-30	27-30#	27-34	27-34#	27-39	27-39#	27-51	27-51#	27-56	27-56#	27-61	27-61#	27-65	27-65#
	27-69	27-69#	27-73	27-73#	27-77	27-77#	27-81	27-81#	67-31	67-31#	81-10	81-10#	82-18	82-18#
	95-5	95-5#	96-1	96-1#	99-38	99-38#	99-45	99-45#	100-8	100-8#	101-8	101-8#	110-10	110-10#
	111-8	111-8#	112-16	112-16#	112-18	112-18#	112-28	112-28#	112-41	112-41#	112-44	112-44#	112-47	112-47#
	112-51	112-51#	112-57	112-57#	117-3	117-3#	118-3	118-3#	119-3	119-3#	121-39	121-39#	121-50	121-50#
	123-12	123-12#												
M\$SVC	27-15	27-15#	27-16	27-16#	27-17	27-17#	27-20	27-20#	27-21	27-21#	27-22	27-22#	27-25	27-25#
	27-26	27-26#	27-27	27-27#	27-28	27-28#	27-31	27-31#	27-32	27-32#	27-35	27-35#	27-36	27-36#
	27-37	27-37#	27-40	27-40#	27-41	27-41#	27-42	27-42#	27-45	27-45#	27-49	27-49#	27-52	27-52#
	27-53	27-53#	27-54	27-54#	27-57	27-57#	27-58	27-58#	27-59	27-59#	27-62	27-62#	27-63	27-63#
	27-66	27-66#	27-67	27-67#	27-70	27-70#	27-71	27-71#	27-74	27-74#	27-75	27-75#	27-78	27-78#
	27-79	27-79#	27-82	27-82#	27-83	27-83#	28-7	28-8	28-8#	32-20	32-21	32-21#	34-22	34-40
	34-43	34-43#	34-54	34-54#	35-11	35-18	35-25	36-6	41-32	41-32#	41-34	41-34#	41-42	41-42#
	41-43	41-43#	41-44	41-44#	41-45	41-45#	41-53	41-53#	41-59	41-59#	41-62	41-62#	41-66	41-66#
	41-113	41-113#	49-28	49-28#	49-33	49-33#	49-34	49-34#	51-27	51-27#	51-60	51-60#	51-63	51-63#
	52-41	52-41#	52-44	52-44#	53-21	53-21#	55-18	55-18#	56-6	56-6#	56-14	56-14#	56-43	56-43#
	56-56	56-56#	56-6	56-6#	58-13	58-13#	58-20	58-20#	59-24	61-59	61-59#	61-69	61-69#	64-39
	64-39#	64-43	64-43#	65-4	65-4#	65-13	65-13#	65-30	65-30#	67-27	67-27#	67-34	67-34#	67-37
	67-37#	67-48	67-48#	68-32	68-32#	71-3	75-28	75-28#	75-36	75-44	77-17	77-17#	77-36	77-36#
	85-18	85-37	86-13	86-13#	86-24	86-24#	86-27	86-27#	86-30	87-44	88-27	88-27#	88-36	89-5
	92-12	92-12#	93-21	93-21#	93-24	93-24#	93-26	93-26#	93-31	93-31#	93-41	93-41#	93-76	93-76#
	93-83	93-83#	93-89	93-89#	93-99	93-99#	93-102	93-102#	94-1	94-2	94-2#	94-3	94-4	94-4#
	97-10	97-10#	97-11	97-11#	98-23	98-23#	98-24	98-24#	99-59	99-59#	99-70	99-70#	99-71	99-71#
	99-74	99-74#	99-75	99-75#	99-88	99-88#	99-107	99-107#	99-108	99-108#	99-118#	99-156	99-156#	101-63
	101-63#	101-66	101-66#	101-69	101-69#	101-73	101-73#	101-79	101-79#	101-88	101-88#	101-91	101-91#	101-114
	101-114#	101-122	101-123	101-123#	102-1	102-1#	102-3	102-3#	102-13	102-13#	102-15	102-15#	102-18	102-18#
	102-24	102-24#	103-10	103-10#	106-5	106-5#	108-12	108-12#	108-21	108-21#	108-30	108-30#	108-32	108-32#
	108-35	108-35#	108-42	108-42#	108-45	108-45#	109-2	109-3	109-3#	109-6	109-7	109-7#	109-11	109-12
	109-12#	109-15	109-15#	110-18	110-18#	111-30	111-30#	112-28	112-28#	112-30	112-30#	112-34	112-34#	112-37
	112-40	112-40#	112-41	112-41#	112-43	112-43#	112-44	112-44#	112-46	112-46#	112-47	112-47#	112-50	112-50#
	112-51	112-51#	112-56	112-56#	112-57	112-57#	112-66	112-66#	113-6	113-6#	114-17	114-27	115-11	115-11#
	115-19	116-16	116-16#	116-17	116-17#	116-18	116-18#	116-33	116-33#	116-41	116-44	116-44#	116-55	116-55#
	116-61	116-63	116-63#	116-67	116-69	116-69#	116-72	116-72#	116-77	116-77#	117-24	117-24#	118-14	118-14#
	119-12	119-12#	119-19	119-19#	119-24	119-24#	119-36	119-36#	119-64	119-64#	119-68	119-68#	119-76	119-76#
	119-113	119-113#	119-116	119-116#	119-117	119-117#	120-12	120-12#	120-14	120-14#	120-23	120-23#	120-29	120-29#
	120-32	120-32#	120-33	120-33#	120-36	120-36#	120-41	120-41#	120-50	120-50#	120-51	120-51#	120-54	120-54#
	120-57	120-57#	120-59	120-59#	120-77	120-77#	120-78	120-78#	120-79	120-79#	120-80	120-80#	120-81	120-81#
	120-82	120-82#	120-83	120-83#	120-121	120-121#	120-127	120-127#	120-131	120-131#	120-146	120-146#	120-150	120-150#
	120-156	120-156#	120-160	120-160#	120-166	120-166#	120-180	120-180#	120-184	120-184#	120-189	120-189#		

MSTLAB	27-15#	27-16#	27-17#	27-20#	27-21#	27-22#	27-25#	27-26#	27-27#	27-28#	27-31#	27-32#	27-35#	27-36#
	27-37#	27-40#	27-41#	27-42#	27-45#	27-49#	27-52#	27-53#	27-54#	27-57#	27-58#	27-59#	27-62#	27-63#
	27-66#	27-67#	27-70#	27-71#	27-74#	27-75#	27-78#	27-79#	27-82#	27-83#	28-7#	28-8#	32-20#	32-21#
	34-22#	34-40#	34-43#	34-54#	35-11#	35-18#	35-25#	36-6#	41-32#	41-34#	41-42#	41-43#	41-44#	41-45#
	41-53#	41-59#	41-62#	41-66#	41-113#	49-28#	49-33#	49-34#	51-27#	51-60#	51-63#	52-41#	52-44#	53-21#
	55-18#	56-6#	56-14#	56-43#	56-56#	56-64#	58-13#	58-20#	59-24#	61-59#	61-69#	64-39#	64-43#	65-4#
	65-13#	65-30#	67-27#	67-34#	67-37#	67-48#	68-32#	71-3#	75-28#	75-36#	75-44#	77-17#	77-36#	85-18#
	85-37#	86-13#	86-24#	86-27#	86-30#	87-44#	88-27#	88-36#	89-5#	92-12#	93-21#	93-24#	93-26#	93-31#
	93-41#	93-76#	93-83#	93-89#	93-99#	93-102#	94-1#	94-2#	94-3#	94-4#	97-10#	97-11#	98-23#	98-24#
	99-59#	99-70#	99-71#	99-74#	99-75#	99-88#	99-107#	99-108#	99-156#	101-63#	101-66#	101-69#	101-73#	101-79#
	101-88#	101-91#	101-114#	101-122#	101-123#	102-1#	102-3#	102-13#	102-15#	102-18#	102-24#	103-10#	106-5#	108-12#
	108-21#	108-30#	108-32#	108-35#	108-42#	108-45#	109-2#	109-3#	109-6#	109-7#	109-11#	109-12#	109-15#	110-18#
	111-30#	112-28#	112-30#	112-34#	112-37#	112-40#	112-41#	112-43#	112-44#	112-46#	112-47#	112-50#	112-51#	112-56#
	112-57#	112-66#	113-6#	114-17#	114-27#	115-11#	115-19#	116-16#	116-17#	116-18#	116-33#	116-41#	116-44#	116-55#
	116-61#	116-63#	116-67#	116-69#	116-72#	116-77#	117-24#	118-14#	119-12#	119-19#	119-24#	119-36#	119-64#	119-68#
	119-76#	119-113#	119-116#	119-117#	120-12#	120-14#	120-23#	120-29#	120-32#	120-33#	120-36#	120-41#	120-50#	120-51#
	120-54#	120-57#	120-59#	120-77#	120-78#	120-79#	120-80#	120-81#	120-82#	120-83#	120-121#	120-127#	120-131#	120-146#
	120-150#	120-156#	120-160#	120-166#	120-180#	120-184#	120-189#							
MSTSTL	27-15	27-15#	27-16	27-16#	27-17	27-17#	27-20	27-20#	27-21	27-21#	27-22	27-22#	27-25	27-25#
	27-26	27-26#	27-27	27-27#	27-28	27-28#	27-31	27-31#	27-32	27-32#	27-35	27-35#	27-36	27-36#
	27-37	27-37#	27-40	27-40#	27-41	27-41#	27-42	27-42#	27-45	27-45#	27-49	27-49#	27-52	27-52#
	27-53	27-53#	27-54	27-54#	27-57	27-57#	27-58	27-58#	27-59	27-59#	27-62	27-62#	27-63	27-63#
	27-66	27-66#	27-67	27-67#	27-70	27-70#	27-71	27-71#	27-74	27-74#	27-75	27-75#	27-78	27-78#
	27-79	27-79#	27-82	27-82#	27-83	27-83#	28-7	28-7#	28-7#	28-8	28-8#	32-20	32-20#	32-20#
	32-21	32-21#	34-22	34-22#	34-22#	34-40	34-40#	34-40#	34-43	34-43#	34-54	34-54#	35-11	35-11#
	35-11#	35-18	35-18#	35-18#	35-25	35-25#	35-25#	36-6	36-6#	36-6#	41-32	41-32#	41-34	41-34#
	41-42	41-42#	41-43	41-43#	41-44	41-44#	41-45	41-45#	41-53	41-53#	41-59	41-59#	41-62	41-62#
	41-66	41-66#	41-113	41-113#	49-28	49-28#	49-33	49-33#	49-34	49-34#	51-27	51-27#	51-60	51-60#
	51-63	51-63#	52-41	52-41#	52-44	52-44#	53-21	53-21#	55-18	55-18#	56-6	56-6#	56-14	56-14#
	56-43	56-43#	56-56	56-56#	56-64	56-64#	58-13	58-13#	58-20	58-20#	59-24	59-24#	59-24#	61-59
	61-59#	61-69	61-69#	64-39	64-39#	64-43	64-43#	65-4	65-4#	65-13	65-13#	65-30	65-30#	67-27
	67-27#	67-34	67-34#	67-37	67-37#	67-48	67-48#	68-32	68-32#	71-3	71-3#	71-3#	75-28	75-28#
	75-36	75-36#	75-36#	75-44	75-44#	75-44#	77-17	77-17#	77-36	77-36#	85-18	85-18#	85-18#	85-37
	85-37#	85-37#	86-13	86-13#	86-24	86-24#	86-27	86-27#	86-30	86-30#	86-30#	87-44	87-44#	87-44#
	88-27	88-27#	88-35	88-36#	88-36#	89-5	89-5#	89-5#	92-12	92-12#	93-21	93-21#	93-24	93-24#
	93-26	93-26#	93-31	93-31#	93-41	93-41#	93-76	93-76#	93-83	93-83#	93-89	93-89#	93-99	93-99#
	93-102	93-102#	94-1	94-1#	94-1#	94-2	94-2#	94-3	94-3#	94-3#	94-4	94-4#	97-10	97-10#
	97-11	97-11#	98-23	98-23#	98-24	98-24#	99-59	99-59#	99-70	99-70#	99-71	99-71#	99-74	99-74#
	99-75	99-75#	99-88	99-88#	99-107	99-107#	99-108	99-108#	99-156	99-156#	101-63	101-63#	101-66	101-66#
	101-69	101-69#	101-73	101-73#	101-79	101-79#	101-88	101-88#	101-91	101-91#	101-114	101-114#	101-122	101-122#
	101-122#	101-123	101-123#	102-1	102-1#	102-3	102-3#	102-13	102-13#	102-15	102-15#	102-18	102-18#	102-24
	102-24#	103-10	103-10#	106-5	106-5#	108-12	108-12#	108-21	108-21#	108-30	108-30#	108-32	108-32#	108-35
	108-35#	108-42	108-42#	108-45	108-45#	109-2	109-2#	109-2#	109-3	109-3#	109-6	109-6#	109-6#	109-7
	109-7#	109-11	109-11#	109-11#	109-12	109-12#	109-15	109-15#	110-18	110-18#	111-30	111-30#	112-28	112-28#
	112-30	112-30#	112-34	112-34#	112-37	112-37#	112-37#	112-40	112-40#	112-41	112-41#	112-43	112-43#	112-44
	112-44#	112-46	112-46#	112-47	112-47#	112-50	112-50#	112-51	112-51#	112-56	112-56#	112-57	112-57#	112-66
	112-66#	113-6	113-6#	114-17	114-17#	114-17#	114-27	114-27#	114-27#	115-11	115-11#	115-19	115-19#	115-19#
	116-16	116-16#	116-17	116-17#	116-18	116-18#	116-33	116-33#	116-41	116-41#	116-41#	116-44	116-44#	116-55
	116-55#	116-61	116-61#	116-61#	116-63	116-63#	116-67	116-67#	116-67#	116-69	116-69#	116-72	116-72#	116-77
	116-77#	117-24	117-24#	118-14	118-14#	119-12	119-12#	119-19	119-19#	119-24	119-24#	119-36	119-36#	119-64
	119-64#	119-68	119-68#	119-76	119-76#	119-113	119-113#	119-116	119-116#	119-117	119-117#	120-12	120-12#	120-14
	120-14#	120-23	120-23#	120-29	120-29#	120-32	120-32#	120-33	120-33#	120-36	120-36#	120-41	120-41#	120-50
	120-50#	120-51	120-51#	120-54	120-54#	120-57	120-57#	120-59	120-59#	120-77	120-77#	120-78	120-78#	120-79
	120-79#	120-80	120-80#	120-81	120-81#	120-82	120-82#	120-83	120-83#	120-121	120-121#	120-127	120-127#	120-131
	120-131#	120-146	120-146#	120-150	120-150#	120-156	120-156#	120-160	120-160#	120-166	120-166#	120-180	120-180#	120-184
	120-184#	120-189	120-189#											
M\$WORD	5-77	5-77#	6-8	6-8	6-8	6-8	6-8	6-8#	28-7	28-7	28-7	28-7#	32-20	32-20

